

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: «Бот для проведення тендерів з продажу агрокультур»

Виконав:

студент IV курсу, групи ІО-64
Жуковський Антон Леонтійович

Керівник:

Старший викладач, к.т.н.
Габінет Артем Вікторович

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н.
Сімоненко Валерій Павлович

Рецензент:

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій СТИПЕНКО

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проєкт студенту
Жуковському Антону Леонтійовичу

1. Тема проєкту «Бот для проведення тендерів з продажу агрокультур», керівник проєкту Габінет Артем Вікторович, старший викладач, к.т.н., затверджені наказом по університету від «07» травня 2020 р. № 1081-с

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту див. технічне завдання

4. Зміст пояснювальної записки

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) схема функціональна, схема принципова, схема структурна

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко В.П., професор, д.т.н.		

7. Дата видачі завдання _____

* Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проєкту.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Відмітки про виконання
1	<i>Затвердження теми роботи</i>	01.09.2019	
2	<i>Вивчення та аналіз завдання</i>	02.09.2019-02.02.2020	
3	<i>Розробка архітектури та загальної структури програми</i>	03.02.2020-03.03.2020	
4	<i>Розробка структур окремих інтерфейсів програми</i>	04.03.2020-15.03.2020	
5	<i>Програмна реалізація</i>	16.03.2020-12.04.2020	
6	<i>Оформлення пояснювальної записки</i>	13.04.2020-17.05.2020	
7	<i>Захист програмного продукту</i>		
8	<i>Передзахист</i>	26.05.2020	
9	<i>Захист</i>		

Студент

Антон ЖУКОВСЬКИЙ

Керівник

Артем ГАБІНЕТ

Анотація

У бакалаврській дипломній роботі було реалізовано систему для проведення тендерів з продажу агрокультур, використовуючи Telegram Bot API. Розроблена система дозволяє агрокомпанії зручно проводити тендери всередині месенджера Telegram.

Програмний продукт був реалізований на мові програмування Python, у середовищі розробки PyCharm.

Annotation

In the bachelor's thesis, a system for conducting tenders for the sale of crops was implemented using the Telegram Bot API. The developed system allows agricultural companies to conveniently conduct tenders within the Telegram messenger.

The software product was implemented in the Python programming language, in the PyCharm development environment.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467100.002 ТЗ	Бот для проведення тендерів з продажу агрокультур. Технічне завдання	4	
3	A4	ІАЛЦ.467100.003 ПЗ	Бот для проведення тендерів з продажу агрокультур. Пояснювальна записка	58	
4	A4	ІАЛЦ.467100.004 А1	Схема структурна	1	
5	A4	ІАЛЦ.467100.005 А2	Схема функціональна	1	
6	A4	ІАЛЦ.467100.006 А3	Схема принципова	1	

					ІАЛЦ.467100.001 ВП						
Зм.	Арк.	№ докум.	Підпис	Дата							
Розробив		Жуковський А.Л.			Бот для проведення тендерів з продажу агрокультур			Літ.	Аркуш	Аркушів	
Перевірив		Габінет А.В.								1	1
Реценз.								НТУУ «КПІ», ФІОТ, ІО-64			
Н. Контр.		Сімоненко В.П.									
Затв.		Стіренко С.Г.									
Відомість дипломного проекту											

Технічне завдання до дипломного проєкту

на тему: «Бот для проведення тендерів з продажу агрокультур»

Київ – 2020

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	3
5.1. Вимоги до програмного продукту, що розробляється	3
5.2. Вимоги до програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.467100.002 ТЗ						
Зм.	Арк.	№ докум.	Підпис	Дата	Бот для проведення тендерів з продажу агрокультур Технічне завдання	Літ.		Аркуш		Аркушів	
Розробив		Жуковський А.Л.							1	4	
Перевірів		Габінет А.В.									
Реценз.											
Н. Контр.		Сімоненко В.П.									
Затв.		Стіренко С.Г.				НТУУ «КПІ», ФІОТ, ІО-64					

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дане технічне завдання розповсюджується на розробку чат-боту для проведення тендерів з продажу агрокультур.

Область застосування: альтернатива іншим системам для проведення тендерів.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки служить завдання на виконання розробки чат-боту для проведення тендерів з продажу агрокультур, затвердженою кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний Інститут ім. Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проєкту є розробка зручної системи інтегрованої в чат-бот для проведення тендерів з продажу агрокультур.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами для розробки служать науково-технічна література з комп'ютерних технологій, публікації в періодичних виданнях, публікації в Інтернеті за даним питанням.

					ІАЛЦ.467100.003 ПЗ	Арк.
						2
Зм.	Арк.	№ докум.				

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- Можливість створення, видалення, закриття тендерів;
- Можливість перегляду та взаємодії з тендерами;
- Дублювання закритих тендерів в Google Sheets.

5.2. Вимоги до програмного забезпечення серверу

- Операційна система Windows, Linux;
- БД MongoDB;
- Мова програмування Python.

					ІАЛЦ.467100.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.				

6. ЕТАПИ РОЗРОБКИ

	Дата
Затвердження теми роботи	01.09.2020
Вивчення та аналіз завдання	02.09.2019-02.02.2020
Розробка архітектури та загальної структури програми	03.02.2020-03.03.2020
Розробка структур окремих інтерфейсів програми	04.03.2020-15.03.2020
Програмна реалізація	16.03.2020-12.04.2020
Оформлення пояснювальної записки	13.04.2020-17.05.2020
Захист програмного продукту	
Передзахист	26.05.2020
Захист	

Пояснювальна записка

до дипломного проєкту

на тему: «Бот для проведення тендерів з продажу агрокультур»

Київ – 2020

ЗМІСТ

ЗМІСТ	1
ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ	3
ВСТУП.....	4
РОЗДІЛ 1.....	6
ЧАТ-БОТИ ЯК ВАЖЛИВА СКЛАДОВА СПІЛКУВАННЯ У МЕСЕНДЖЕРАХ.....	6
1.1.Світові інформаційні тенденції.....	6
1.2 Месенджери	7
1.3 Опис та класифікація чат-ботів.....	12
1.4 Створення чат-ботів.	15
1.5 Приклади чат ботів	16
1.5.1 Чат-бот по COVID-19 Лікар Іван Маск	16
1.5.2 Сервіс державних даних OpenDataUA	20
1.5.3 Моніторинг відправлень «Нова Пошта»	21
ВИСНОВКИ ДО РОЗДІЛУ 1	23
РОЗДІЛ 2.....	24
ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ	24
2.1 Обрані технології розробки	24
2.1.1 Telegram Bot API.....	24
2.1.2 Python та бібліотека AIOgram	25
2.1.3 MongoDB.....	28

					<i>ІАЛЦ.467100.003 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розробив		Жуковський А.Л.			Бот для проведення тендерів з продажу агрокультур Пояснювальна записка	Літ.	Аркуш
Перевірів		Габінет А.В.					1
Реценз.							58
Н. Контр.		Сімоненко В.П.				НТУУ «КПІ», ФІОТ, ІО-64	
Затв.		Стіренко С.Г.					

2.4 Середовище розробки JetBrains PyCharm	31
ВИСНОВОК ДО РОЗДІЛУ 2	36
РОЗДІЛ 3	37
РОЗРОБКА ЧАТ-БОТУ TELEGRAM	37
3.1 Отримання токена	37
3.2 Організація структури файлів бота.....	38
3.3 Хід розробки	39
3.3.1 Розробка структури БД.....	39
3.3.2 Індивідуальні налаштування бота.....	42
3.3.3 Реєстрація в боті	44
3.3.4 Опис інтерфейсу адмін-панелі	45
3.3.5 Опис інтерфейсу клієнта	48
3.3.6 Опис алгоритму проведення торгів	48
3.3.8 Функція експорту закритих тендерів в Google Sheets	49
3.5 Розміщення та запуск проєкту на сервері	50
3.5.1 Встановлення Python	50
3.5.2 Встановлення необхідних бібліотек	52
3.5.3 Реалізація автоматичного запуску бота при перезавантаженні сервера.....	53
ВИСНОВОК ДО РОЗДІЛУ 3	56
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	58

ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ

ОС	Операційна система
Фреймворк	Інфраструктура програмних рішень, що полегшує розробку складних систем.
БД	база даних
СМОП	Система Миттєвого Обміну Повідомленнями
ЄДРПОУ	Єдиний державний реєстр підприємств та організацій України
API	Прикладний програмний інтерфейс (англ. Application Programming Interface, API)
JSON	Текстовий формат обміну даними між комп'ютерами, який дозволяє описувати об'єкти та інші структури даних (англ. JavaScript Object Notation)
HTTP	(англ. HyperText Transfer Protocol) – протокол, який дозволяє отримувати різні ресурси, наприклад HTML-документи
URL	(англ. Uniform Resource Locator) – система унікальних адрес електронних ресурсів
VPS	Віртуальний виділений сервер(англ. virtual private server)
FSM	Машина станів(англ. finite-state machine)
GS	Google Sheets
IDLE	Integrated Development and Learning Environment
AI	Штучний інтелект

ВСТУП

Мережа Інтернет, яка спочатку була як середовище для передачі інформації, в даний час все більше перебирає на себе функції комунікатора. Глобальна мережа стає особливим середовищем спілкування, яке займає важливе місце в усіх сферах життєдіяльності суспільства. Особливо це стосується сучасного покоління, у якого домінують мобільні пристрої.

На сьогоднішній день проблема віртуального спілкування в мережі стала популярною внаслідок можливостей швидкого доступу до інформації, одночасної роботи в мережі багатьох користувачів, збору та обміну різноманітною інформацією, підтримки навчання і підвищення кваліфікації, комунікацій для вирішення особистих і бізнес-питань з клієнтами і партнерами, проведення всіляких аналітичних досліджень та інших переваг.

Комунікаційні функції Інтернет набувають нових форм, які пов'язані з сучасними технічними можливостями. Особливої актуальності останнім часом набули технології комунікацій, засновані на месенджерах і чат-ботах.

Месенджери – це програми, які дозволяють передавати повідомлення в реальному часі через Інтернет. Еволюція месенджерів привела до того, що в даний час вони можуть відправляти як текстову інформацію, так аудіо і відео повідомлення. Сучасне покоління вважає за краще мобільні пристрої замість настільних комп'ютерів. Його представники не звикли взаємодіяти в Інтернет середовищі за допомогою браузерів і спливаючих меню, а використовують смартфони для невеликої кількості завдань. Дослідження показали, що телефони використовуються для обміну повідомленнями частіше, ніж для інших цілей. Тому компанії прагнуть завоювати увагу онлайн-користувачів і створюють чат-боти, щоб вбудуватися в месенджери.

Одна з приємних особливостей ботів є те, що для взаємодії з ними зі сторони користувача, абсолютно непотрібно мати які-небудь технічні знання або навички програмування. По суті, це нагадує діалог зі звичайною людиною,

					ІАЛЦ.467100.003 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

тільки різниця у тому, що «співрозмовник» розуміє тільки певні команди, на які він і реагує.

На сьогоднішній день, використання ботів Telegram, дає цілий ряд переваг для бізнесу: досить швидко реагують на запити та вирішують потрібні задачі(1-3 секунди); боти можуть швидко інформувати своїх користувачів про різні акції, новинки, спеціальні пропозиції тощо; можна організувати просту техпідтримку або консультацію; це комунікація напрямку з потенційними клієнтами; відносна дешевизна розробки; синхронізація між пристроями – чат з ботом буде доступний будь-де(як на комп'ютері, так і на мобільному пристрої чи планшеті), потрібно просто авторизуватися за своїм обліковим засобом. Боти можуть працювати або з конкретними сайтами чи інтернет-магазинами, або парсити інформацію з якогось агрегатора.

Але на превеликий жаль ми маємо зрозуміти що можливості ботів обмежені, якщо порівнювати їх з мобільними додатками або веб-сайтами.

					ІАЛЦ.467100.003 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

ЧАТ-БОТИ ЯК ВАЖЛИВА СКЛАДОВА СПІЛКУВАННЯ У МЕСЕНДЖЕРАХ

1.1. Світові інформаційні тенденції

В технологічному просторі проявляються основних два напрями: з однієї сторони, об'єм загальнодоступної інформації збільшується в геометричній прогресії, з іншого боку, самі дані фізично стискаються. При чому другий напрям розповсюджується як на передачу, так і зберігання даних. Всі бібліотеки світу зараз можна стиснути до одного електронного ресурсу, спілкування в соц. мережах – до постів, сторіс та лайків, а взаємодія між всіма користувачами мережі, між клієнтами і компаніями зводиться до взаємодії через чат-боти.

Основне призначення чат-ботів це заміна мобільних програм. Не потрібно нічого скачувати, всі потрібні тобі додатки в розміщені в одному месенджері. Чат-боти найчастіше використовують для прискорення процесу взаємодії між клієнтами та бізнесом. Часто ботів використовують для швидкого доступу до відповідей на актуальні питання, вирішення різних проблем, та задоволення потреб користувача. Flurry Analytics було проведене дослідження, яке показало що потреби в месенджерах(програм для обміну повідомленнями між користувачами) продовжують зростати.[4]. В останній час люди не мають бажання встановлювати багато програм, біля 72% людей використовують не більше 7 мобільних додатків протягом дня[5] .

Тому найкращий спосіб на даний момент, є інтеграція всіх можливих сервісів в програму, з якою користувач вже знайомий. Саме таку можливість надають месенджери додавши можливість створення функціональних чат-ботів. Нинішні платформи в месенджерах надають можливість реалізувати більшість потрібних функціоналів, таких як замовлення їжі, бронювання білетів, готелів, обмін валют та інші. Як перевага так і недолік полягає в шаблонному інтерфейсі без можливості кастомізації, користувачу не потрібно звикати до нового інтерфейсу, адже він відносно схожий у всіх чат-ботах на

					ІАЛЦ.467100.003 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

платформі, але розробнику потрібно підлаштовуватись під обмежений інтерфейс.

1.2 Месенджери

В даний час популярність месенджерів як засобів спілкування незмінно зростає. Компанії, сім'ї, друзі щодня користуються можливостями обміну повідомленнями та медіаконтенту на відстані. так само варто відзначити зростання популярності такого виду програмних продуктів як чат-боти, які працюють на платформах месенджерів. Причина такої зростаючої популярності засобів для обміну повідомлення зумовлена змінами в області мобільного інтернету: тепер він має високу швидкість, став набагато доступніший у фінансовому плані, а також смартфони стали доступнішими.

Усі месенджери на сьогоднішній день мають багато ідентичних функцій, а саме: миттєва відправка повідомлень, аудіо фото та відео файлів, можливість відео чи аудіо спілкування.

На сьогоднішній день існує багато різноманітних месенджерів, черед яких я визначив найпопулярніші, та відобразив на таблиці їх переваги та недоліки(табл. 1.1).

Табл. 1.1 Переваги та недоліки месенджерів

Назва	Переваги	Недоліки
WhatsApp	Доступ по протоколу HTTPS	Істотне зниження якості переданих медіафайлів
	Групові чати до 256 учасників	
	Доступ до чату в режимі офлайн	Велика затримка в відеочаті
	Зручні голосові та відеозвінки	

Табл. 1.1(продовження) Переваги та недоліки месенджерів

Viber	Вбудовані служби для оплати в мережі Інтернет	Наявність великої кількості реклами
	PUSH-сповіщення в режимі офлайн	Наявність вбудованих покупок
	Наявність публік-акаунтів	
Skype	Зручні групові дзвінки і відеоконференції	Сильна залежність від швидкості Інтернет з'єднання
	Демонстрація екрану під час дзвінка або відеотрансляції	Часто нестабільне з'єднання
		Відсутність шифрування
iMessage	Великий набір стандартних анімацій чату	Велика затримка доставки повідомлень
	Висока ступінь інтеграції зі сторонніми додатками Apple	Працює тільки на системі iOS
Hangouts	Особливий протокол передачі голосу Google Voice	Інтуїтивно не зрозумілий інтерфейс
	Історія повідомлень зберігається на віддаленому сервері	Сильне стиснення медіафайлів
	Інтегровані сервіси Google+ і Gmail	

Табл. 1.1(продовження) Переваги та недоліки месенджерів

Facebook Messenger	Підтримка ботів	Конфіденційність даних
	Шифрування повідомлень	
	Функція «Секретний чат»	Неможливість редагування повідомлень
	Самознищення обраних повідомлень через зазначений час	
Telegram	Найбільш зручна платформа для чат-ботів	Відсутні відеодзвінки
	Спеціально створений протокол шифрування MTProto	Десктопні версії не підтримують секретні чати
	Секретні чати	

Особливої уваги заслуговує месенджер Telegram.

Telegram — це програма яка доступна на всіх операційних системах, яка була написана на мові програмування C++, і дозволяє обмінюватись між користувачами повідомленнями а також файлами будь-яких форматів. Він використовує серверну частину яка була спеціально розроблена. Працює він на серверах США та Німеччини

Telegram має такі переваги:

- приватність - всі чати шифруються, а повідомлення знищуються через певний час;
- швидкість – велика швидкість доставки повідомлення, вища ніж у конкурентів;
- розподіл – Telegram сервери розміщені по всьому світу, це зменшує кількість «падінь»;

- телеграм використовує відкритий протокол MTProto і API, яке є безкоштовним для всіх користувачів;
- також відсутні будь які підписки та зовсім відсутня будь-яка реклама;
- немає ніяких обмежень на розмір вілправлених повідомлень і медіафайлів.

Один з суттєвих мінусів є відсутність в месенджері відеодзвінків.

Месенджер Telegram заснували два брати по імені Микола та Павло Дурови. Перший відповідає за технологічну складову проєкту, а другий за фінанси та ідеологію проєкту

Відмінними особливостями месенджеру є наявність секретних чатів та розроблений спеціальний протокол шифрування MTProto.

Спеціально для людей, які хочуть максимально безпечно спілкуватись у мережі було створено секретні чати. Окрім використання спеціально розробленого протоколу, повідомлення шифруються ключами по принципу від пристрою до пристрою, тому ніхто не може перехопити і розшифрувати повідомлення, навіть співробітники Telegram. Особливості секретних чатів це те, що повідомлення які надіслані в секретних чатах не можливо переслати, якщо був зроблений, співрозмовник буде повідомлений про це. Але є й недолік в цих чатах, чат можна відкрити тільки з того пристрою, з якого він був створений.

Telegram месенджер підтримує всі популярні операційні системи, такі як IOS, Windows, Linux, Android, MacOS та інші. Перевагою також є те, що месенджер підтримує як мінімум українську, англійську, російську, польську та німецьку мови. Також цей список активно поповнюється.

На рисунках 1.1 і 1.2 зображено приклади інтерфейсу месенджера на таких платформах як IOS 13 та Windows 10.

					ІАЛЦ.467100.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

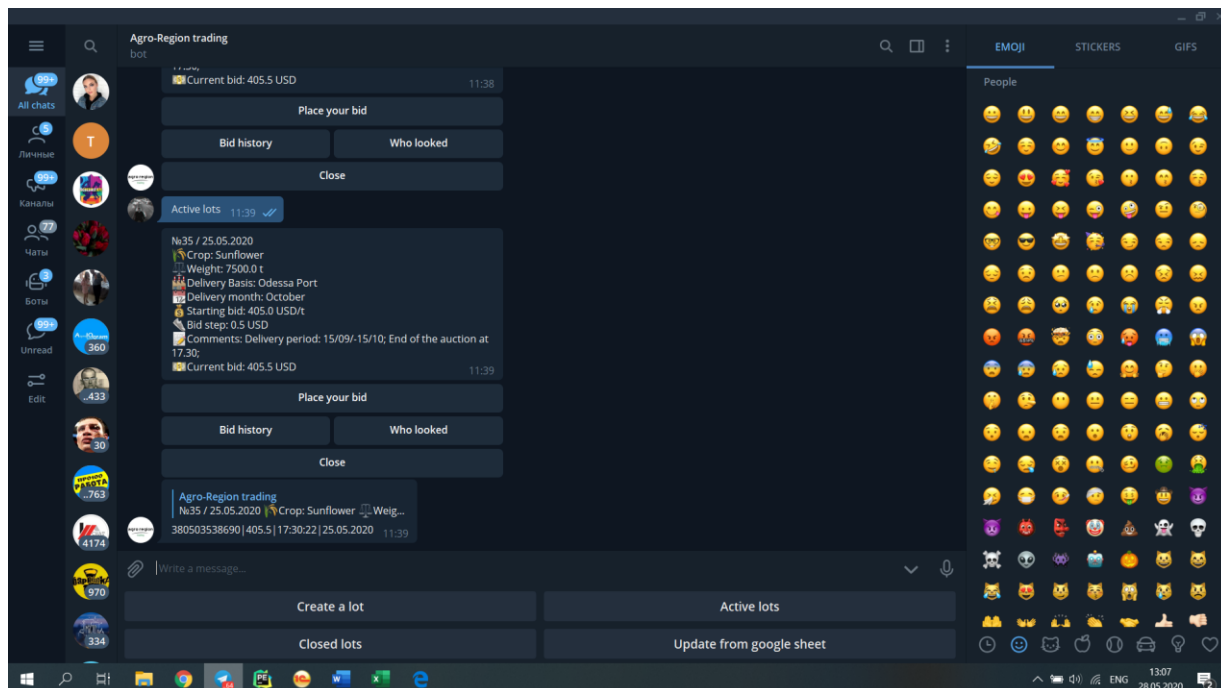


Рис. 1.1 Інтерфейс Telegram на Windows 10

На рисунках ми можемо побачити що інтерфейс у Telegram дійсно максимально зрозумілий та зручний. Також можна помітити повну відсутність будь якої реклами, це як на мене одна з найбільших переваг Telegram серед інших месенджерів.

Також в одному з останніх оновлень був доданий поділ всіх чатів по категоріям, при чому можна використовувати як стандартні категорії, так і створювати їх самостійно. Це виводить Telegram ще на крок вперед серед інших месенджерів, адже подібної функції немає ні в одному месенджері.

					ІАЛЦ.467100.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

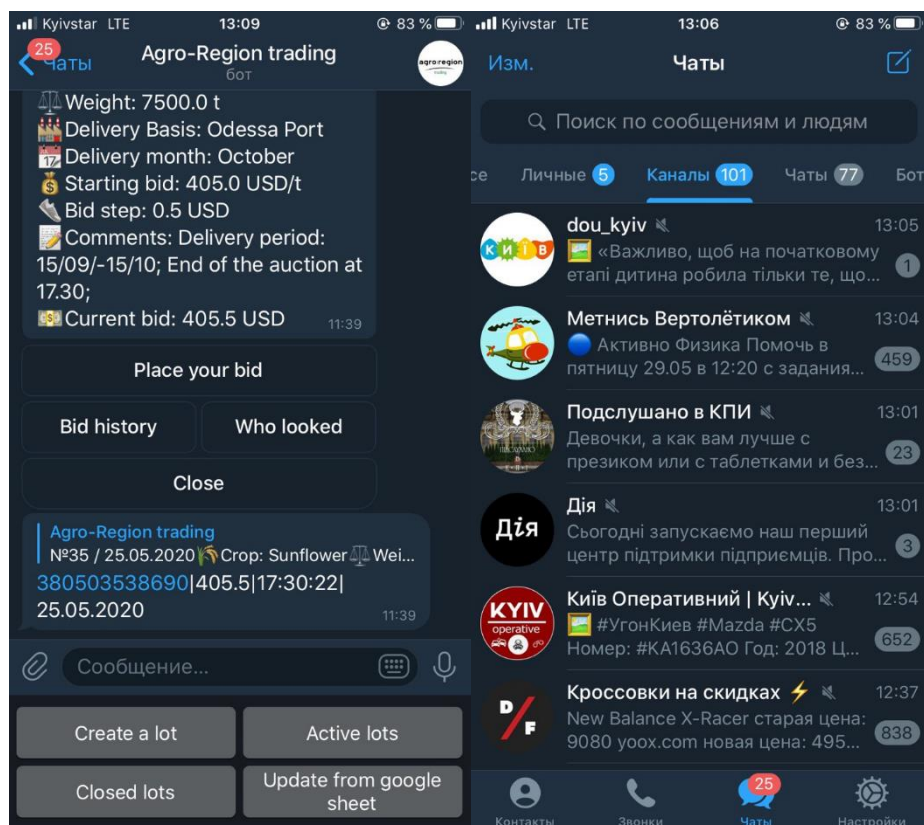


Рис. 1.2 Інтерфейс Telegram на iOS 13

Останнім часом Telegram повідомив що веде розробку відео-дзвінків, що на мою думку, в загальному допоможе йому стати на голову вище конкурентів.

1.3 Опис та класифікація чат-ботів

На даний момент чат-боти в месенджерах є універсальним вирішенням різних потреб і завдань. Бот (англ. Bot) – програма, яка автоматично або за певним розкладом виконує якісь завдання через обмежений платформною інтерфейс. Часто боти використовують для виконання одноманітних задач, таких як відповіді на поширені запитання, передача, повідомлення про різноманітні події тощо. Їх використовують, так як вони мають швидшу реакцію, ніж люди. Це наприклад ігри всередині чат-бота, аукціони тощо. Також часто їх використовують для створення вигляду спілкування з людиною, тоді це рахуються чат-боти. Чат-боти – це програмні платформи, як доступні в месенджерах, і можуть вести себе як людина. Такі програми здатні давати відповіді на запитання, а також задавати їх у відповідь. Вони

					ІАЛЦ.467100.003 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

використовуються в різноманітних відгалуженнях для виконання типових завдань. За даними досліджень компанії Business Insider до 2020 року 80% компаній буде користуватися чат-ботами; 42% учасників опитування вважають, що технології чат-ботів найбільш поліпшать якість обслуговування клієнтів; 48% – вже використовують технологію чат-ботів для таких бізнес-функцій і 40% – планують впровадити деяку форму даної технології до 2020 року [3]. На сьогоднішній день існує багато різноманітних чат-ботів, яких використовують для торгівлі, в сферах охорони здоров'я та фінансів. Навіть в процесі навчання вони стають частим явищем.

Для того щоб зрозуміти сутність, переваги та недоліки чат-ботів, я привів приклад їх класифікації (табл. 1.2)

Табл 1.2 Класифікація чат-ботів

Ознака класифікації	Вид чат-бота
Користувач	Персональний Бізнесовий
Інтерфейс	Кнопковий Текстовий
Призначення	Комунікаційний Функціональний
Принцип роботи	Шаблонний Який навчається

Використовуватись чат-боти можуть по різному: для особистого або ж ділового використання.

Чат-боти для особистого користування використовуються як помічники юзера та виконують різноманітні задачі, такі як прийом медіафайлів, викликів, управління розпорядком користувача тощо.

Чат-боти для ділового користування розробляються для взаємодії бізнесу та клієнтів, вони в процесі діалогу виконують різні завдання, пов'язані з продажом, маркетингом тощо. Також такі боти застосовуються в

різноманітних сферах бізнесу для автоматизації завдань, процесів а також різних аналітичних завдань.

Спілкування з користувачем у чат-бота реалізоване за допомогою кнопок або ж певних команд чи фраз. Спілкування між користувачем і ботом реалізовано так, що користувачу на вибір запропоновані питання, категорії або ж пропозиції, які могли б привернути його інтерес. Користувач за допомогою натискань по кнопкам, передає потрібні команди боту, який в свою чергу реагує на них, і дає потрібну відповідь. Такі чат-боти дуже схожі на мобільні програми, які відрізняються лише тим, що інтерфейс обмежений можливостями месенджера. Кнопки та команди для таких чат-ботів прописуються при його розробці.

Головна функція чат-ботів це повністю замінити функціонал мобільної програми. Їх використовують для пришвидшення процесу взаємодії між бізнесом(компаніями) та клієнтами. Найбільш часто чат-ботів використовують для відповідей на найбільш поширені питання.

Чат-боти можна поділити на два типи за принципом спілкування: ті, які працюють за шаблоном, та ті, які можуть в процесі спілкування навчатись. Боти які працюють за шаблонами не розуміють звичайне спілкування. Принцип спілкування в таких ботах наступний: є зарання створений шаблон, а самий скрипт представляє з себе дерево сценаріїв, в якому кнопка відкриває новий відповідний сценарій. Такі боти при взаємодії з користувачем розуміють шаблонні слова та реагуються тільки на них. Для кожної фрази або словосполучення в такому боті потрібно прописати необхідну команду . У випадку, коли користувач спілкується з ботом не використовуючи шаблонні слова, то чат-бот не може його зрозуміти, і або не реагує взагалі, або відповідає зарання заготовленими фразами по типу «я тебе не розумію», або ж «звернись до менеджера(оператора)».

Чат-боти, які в процесі спілкування навчаються, розробляються за допомогою AI(штучного інтелекту), використовуючи при побудові діалогу зчитування природньої мови та машинне навчання. Самий принцип таких чат-

					ІАЛЦ.467100.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

ботів побудований на аналізі повідомлень для вдосконалення своїх розмовних навичок. Такі боти розуміють просте спілкування користувача і розуміючи сенс діалогу, влучно відповідають на відповідні запитання.

1.4 Створення чат-ботів.

Швидке зростання популярності ботів породило за собою створення багатьох різноманітних платформ для їх розробки [1; 2; 6]. Можна створити чат-бот з нуля, використовуючи якусь мову програмування. Для розробки чат-ботів в більшості випадків використовують такі мови програмування: Python, PHP, Ruby on rails, JS тощо. Також є багато платформ, які пропонують готові рішення для створення чат-ботів, вони допоможуть без написання коду навчити його розуміти та виконувати потрібні команди.

Для того щоб створити чат-бот потрібно потратити чимало зусиль прогятом всіх етапів розробки. Основні зусилля розробники тратять на навчання чат-бота розумінню контексту звернення користувача (26%) а також побудову логіки діалогу (16%) , написання програмного коду – 27% [4].

Для створення чат-боту потрібно визначити завдання які він повинен буде вирішувати, обрати потрібну платформу, для якої будемо його розробляти (Viber, Telegram, Facebook тощо), обрати сервіс для розбки а також сервіс, де він буде розміщений(VPS сервер).

При створенні можна виділити декілька етапів:

1) Визначити вимоги:

- ознайомитися з предметною областю для чат- бота і визначити завдання, які він буде вирішувати;
- сформуванати вимоги до розробки чат-бота;
- вибрати платформи, через які відбуватиметься спілкування, вибрати сервіс для розробки та сервіс для запуску бота.

2) Розроблення чат-бота:

- створити комунікаційну архітектуру, яка дозволить вибудувати ієрархію контенту для спілкування;

					ІАЛЦ.467100.003 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

- зібрати інформацію, необхідну для повноцінного діалогу, щоб навчити бота розуміти різні варіанти одного і того ж поняття і давати коректні відповіді. (Це можна зробити, провівши опитування серед групи людей цільової аудиторії);
- задати схему ведення діалогу на основі власного сценарію або на основі умов, при яких бот буде відповідати певним чином;
- структурувати контент і змодельовати схему діалогу, детально опрацьовуючи відповіді ботів, щоб спілкування виглядало природним;
- навчити бота, розробивши чіткі структуровані сценарії відповіді на основі великої бази даних для навчання або використовуючи готові навчальні алгоритми для інтерпретації відповідей в разі відсутності такої бази.

Можна зробити висновок, що для розробки чат-бота потрібні спеціальні навички і чимало зусиль і широкого кола професійних навичок.

1.5 Приклади чат ботів

1.5.1 Чат-бот по COVID-19 Лікар Іван Маск

Цей чат-бот був створений в зв'язку з актуальною загрозою COVID-19.

Він призначений повідомити про поточну ситуацію, порадити, як захистити себе і близьких, а також ще декілька інших функцій, про які розповім разом з демонстрацією інтерфейсу нижче.

Після команди /start ми бачимо головне меню на клавіатурі(рис. 1.3).

					ІАЛЦ.467100.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

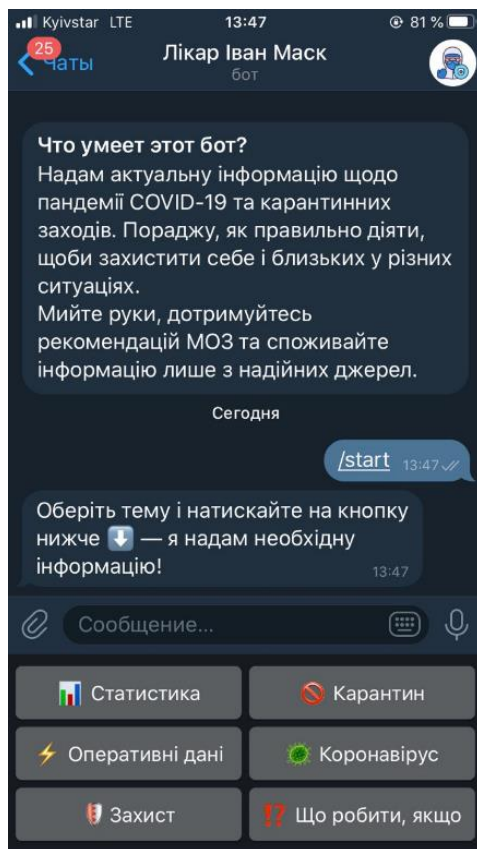


Рис 1.3 Головне меню бота

Кнопка «Статистика» — демонструє дані лабораторно підтверджених випадків зараження COVID-19 станом на поточну дату(рис. 1.4).

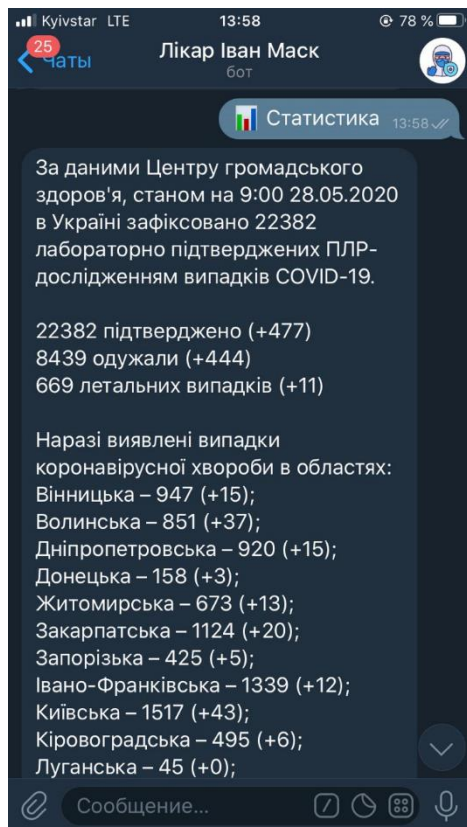


Рис. 1.4 Пункт меню «Статистика»

					ІАЛЦ.467100.003 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

Кнопка «Оперативні дані» — надає актуальну інформацію щодо рівню захворюваності в Україні, обмежень у зв'язку з карантином, усі необхідні контакти та посилання на надійні джерела інформації (рис. 1.5).

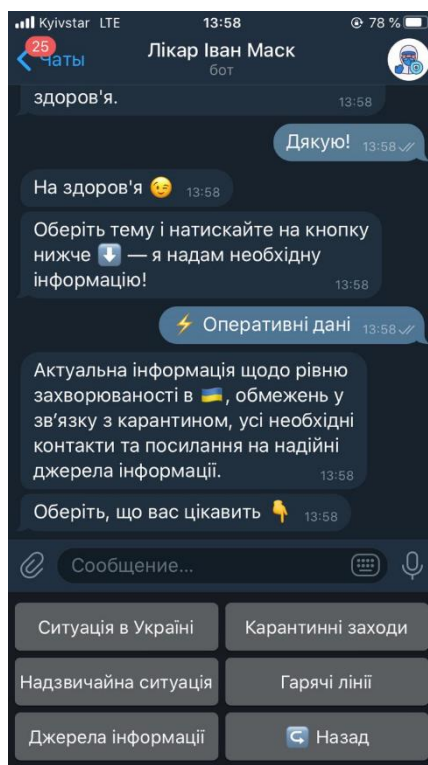


Рис. 1.5 Пункт меню «Оперативні дані»

Кнопка «Карантин» — виводить актуальну інформацію на рахунок карантину, введені пом'якшення та критерії для його послаблення (рис. 6).

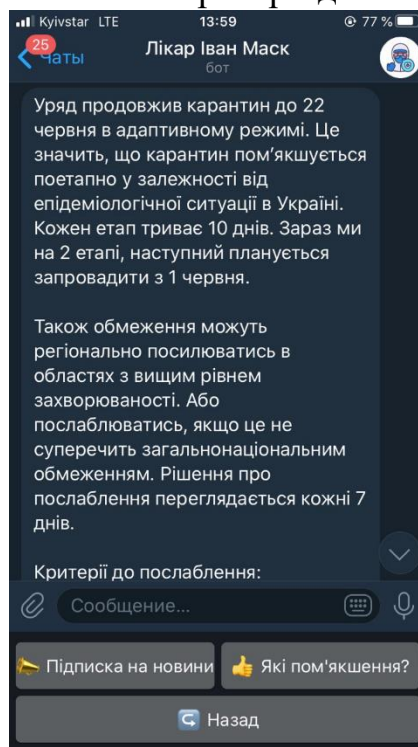


Рис. 1.6 Пункт меню «Карантин»

Зм.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.467100.003 ПЗ

Арк.

18

Кнопка «Коронавірус» — показує що наразі відомо про хворобу COVID-19, також інформацію щодо симптомів, шляхів передачі, лікування, та поширених міфів(рис. 1.7).

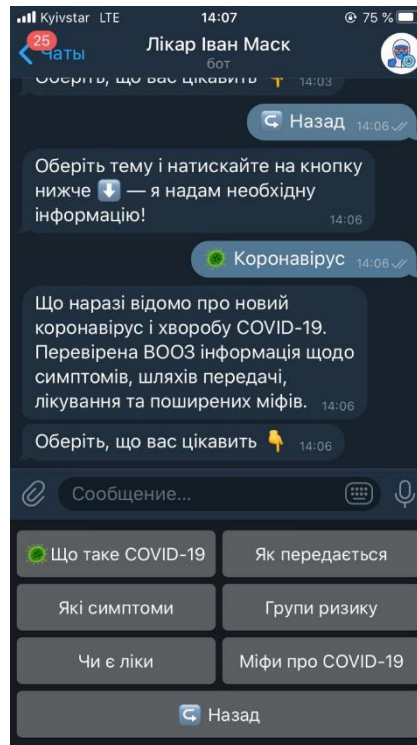


Рис. 1.7 Пункт меню «Коронавірус»

Кнопка «Захист» — повна інформація про те, як захистити себе та близьких від можливого зараження COVID-19(рис. 1.8).

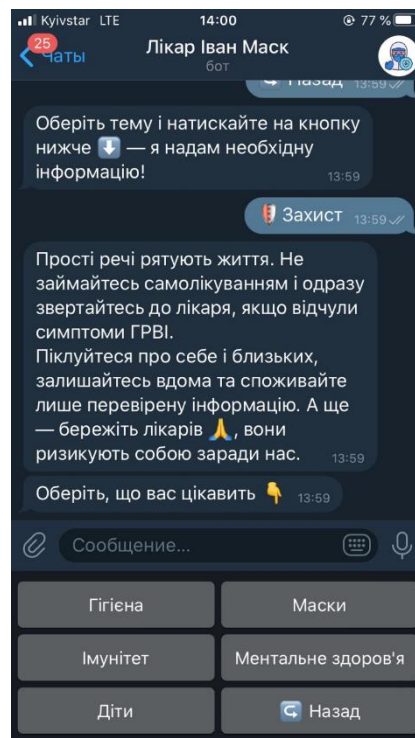


Рис. 1.8 Пункт меню «Захист»

					ІАЛЦ.467100.003 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

Кнопка «Що робити, якщо» — надає рекомендації та алгоритм дій якщо у вас є симптоми ГРВІ, умови проходження тесту, умови для обсервації та самоізоляції(рис. 1.9).

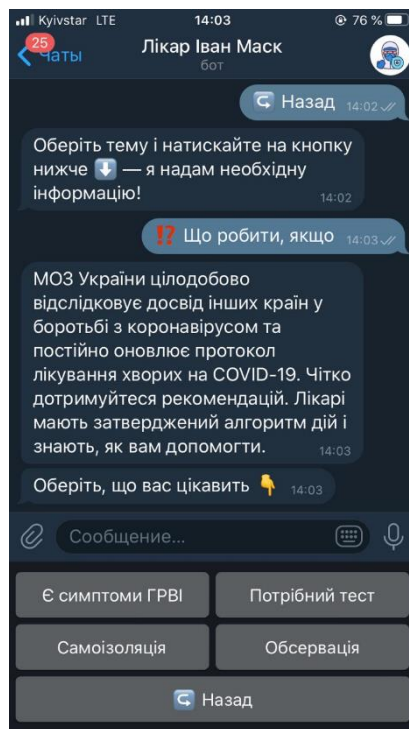


Рис. 1.9 Пункт меню «Що робити, якщо»

1.5.2 Сервіс державних даних OpenDataUA

OpenDataUA – це чат бот для пошуку та моніторингу відкритих державних даних компаній, громадян та авто. Пошук ведеться у всіх реєстрах за кодом ЄДРПОУ, назвою компанії, ПІБ людини та номером авто. Повний перелік інформації для пошуку:

- код, назва чи адреса компанії
- ПІБ людини українською мовою
- номер паспорта
- податковий номер фізичної особи
- номер авто чи техпаспорта
- номер судової справи

Інтерфейс головного меню зображено на рисунку 1.10.

					ІАЛЦ.467100.003 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

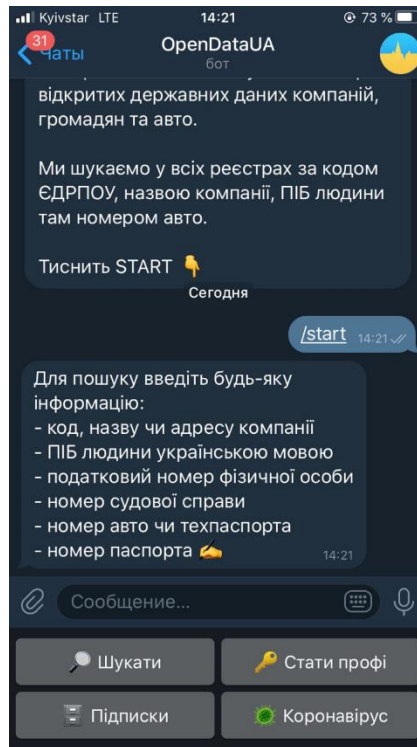


Рис. 1.10 Інтерфейс головного меню

Також як бачимо на рисунку 10 є додаткове меню «Коронавірус», в якому знаходиться актуальна інформація перебігу захворювання COVID-19.

1.5.3 Моніторинг відправлень «Нова Пошта»

Чат-бот дозволяє швидко дізнатись статус ваших відправлень (рис. 1.11), також дізнатись де знаходиться найближче відділення «Нова Пошта» (рис. 1.12).

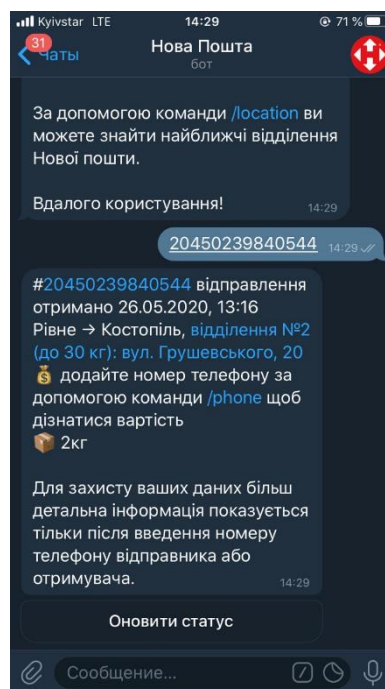


Рис 1.11 Моніторинг статусу відправлення

					ІАЛЦ.467100.003 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

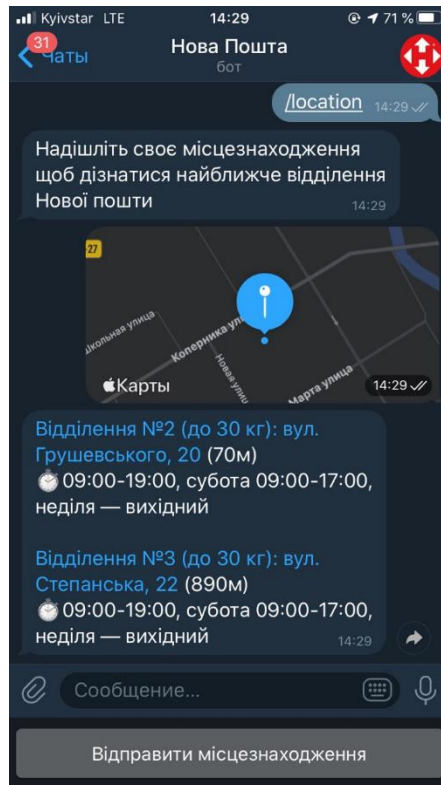


Рис. 1.12 Пошук найближчого відділення

Дізнатись цю інформацію можна також і на офіційному сайті компанії, але набагато швидше та зручніше це робити через чат-бот.

					ІАЛЦ.467100.003 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ ДО РОЗДІЛУ 1

В роботі був проведений аналіз особливостей існуючих рішень для розробки чат-ботів, а саме:

- проведено аналіз найпопулярніших месенджерів;
- описано, що чат-боти на даний момент входять до найбільш перспективних напрямків в майбутньому а також наразі є актуальним для наукових досліджень;
- описана класифікація чат-ботів;
- визначено зміст основних етапів створення чатботів;
- наведено приклади чат-ботів.

					ІАЛЦ.467100.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ

2.1 Обрані технології розробки

2.1.1 Telegram Bot API

Bot API представляє собою HTTP-інтерфейс для роботи з ботами в Telegram. Кожний бот – це спеціальний аккаунт, який створений для оброблення і відправлення повідомлень[9]. Взагалі існує два зовсім різних способів отримання оновлень:

- long pulling – використовуючи цей спосіб ваш додаток самостійно опитує сервера на наявність оновлень з певним інтервалом. За замовчуванням 100мс;
- webhook – використовуючи цей спосіб, оновлення будуть самі приходити на вказаний вами url[9].

Зберігаються оновлення на сервері поки їх не опрацюють, або ж не довше ніж 24 години. Незважаючи на те, який варіант вибрано, результат буде отриманий в форматі JSON.

На рисунку 2.1 зображено принцип взаємодії користувача і чат-бота.

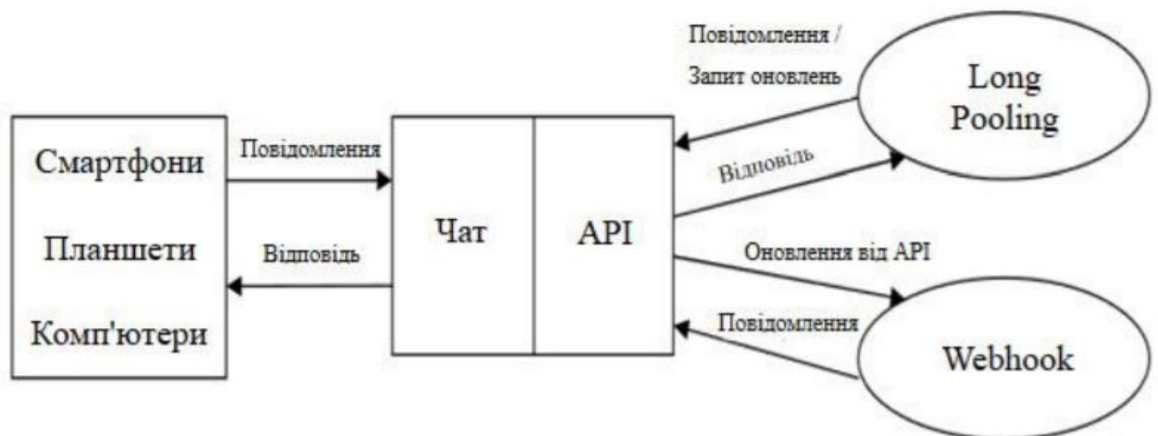


Рис. 2.1 Принцип взаємодії чат-бота і користувача

До Telegram Bot API запити мають бути здійснені через Https в такому вигляді: https://api.telegram.org/bot<token>/НАЗВА_МЕТОДУ[9].

Всі методи (а їх досить багато) діляться на групи:

1. Отримання оновлень і інформації.
2. Робота в чаті.
3. Відправка різних елементів.
4. Робота зі стікерами.
5. Оновлення повідомлень.
6. Режим inline.
7. Платіжний функціонал.
8. Для ігор.

Обмін повідомленнями відбувається у вигляді запитів. У наступній таблиці наведено приклади деяких з них(табл. 2.1).

Табл. 2.1 Приклади декількох методів Telegram Bot API[12]

Назва методу	Дія
getMe	Дозволяє отримати інформацію про користувача
sendMessage	Відправляє повідомлення
sendPhoto	Відправляє фото
sendAudio	Відправляє аудіо
sendDocument	Відправляє документ
sendVideo	Відправляє відео
sendContact	Відправляє контакт
getUpdates	Отримує оновлення з чату

2.1.2 Python та бібліотека AIOgram

Python це мова програмування загального призначення, яка націлена більше на підвищення продуктивності розробника, а ніж самого коду, який пише. Кажучи по простому, написати на Python можливо без проблем практично що завгодно (веб- / настільні додатки, ігри, скрипти автоматизації, системи управління життєзабезпеченням і багато іншого). Більш того, поріг входження низький, а самий код виглядає лаконічно, і його зможе зрозуміти

навіть той, хто ніколи не писав на ньому. За рахунок простоти коду, подальша підтримка програм, написаних на Python, стає легша і приємніша в порівнянні з Java або C ++[10]. А якщо це розглядати з точки зору бізнесу, це веде за собою збільшення продуктивності співробітників, а також значне скорочення витрат.

Безсумнівною перевагою є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першою такою мовою була C, але на різних комп'ютерах її типи даних могли займати відмінну кількість пам'яті, і це призводило до проблем написання переносних програм. Але в мові Python такий недолік відсутній[11].

Так само, важлива риса - розширюваність мови, цьому надається велике значення і, як пише сам автор Гвідо ван Россум, мова була задумана саме як розширювана. Це означає, що зацікавлені програмісти можуть вдосконалити цю мову програмування. Самий інтерпретатор розроблений на мові програмування C і вихідний код є в загальному доступі. Якщо виникне така потреба, то його можна використовувати як вбудовану оболонку, вставивши в свою програму. Або ж, є можливість отримати розширений інтерпретатор, написавши свої доповнення до нього на мові C[11].

Наступна перевага - наявність багатьох програмних модулів і бібліотек, що забезпечують різні додаткові можливості.

Aiogam це досить простий і повністю асинхронний фреймворк для API Telegram Bot, написаний на Python 3.7 з asyncio та aiohttp. Він використовується для пришвидшення, спрощення і мінімізації написаного коду.

Коли говорять про виконання програм, то під «асинхронним виконанням» розуміють таку ситуацію, коли програма не чекає завершення якогось процесу, а продовжує роботу незалежно від нього. Застосування асинхронних механізмів при написанні якоїсь програми означає, що ця програма буде виконуватися швидше, ніж без використання подібних механізмів.

					ІАЛЦ.467100.003 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

Для знайомства з бібліотекою покажу приклад створення просто боту, який буде вітати користувача та повторювати фрази відправлені користувачем.

Створимо файл bot.py в якому імпортуємо всі необхідні модулі і токен боту, а також ініціалізуємо бот і диспетчера(рис. 2.2).

```
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor

from config import TOKEN

bot = Bot(token=TOKEN)
dp = Dispatcher(bot)
```

Рис 2.2 Вміст файлу bot.py

Команда з якої користувач починає спілкуватись із ботом - /start, тому потрібно бота навчити відповідати на цю команду. Створюємо хендлер і оголошуємо всередині функцію відповіді(рис. 2.3).

```
@dp.message_handler(commands=['start'])
async def process_start_command(message: types.Message):
    await message.reply("Привіт! Напиши щось!")
```

Рис 2.3 Реалізація реагування на команду /start

Тепер створимо хендлер для дублювання повідомлень користувача йому ж(рис 2.4).

```
@dp.message_handler()
async def echo_message(msg: types.Message):
    await bot.send_message(msg.from_user.id, msg.text)
```

Рис. 2.4 Код дублювання повідомлень користувача

					ІАЛЦ.467100.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

І тепер для того щоб отримувати повідомлення від Telegram, потрібно використати пуллінг – постійне опитування серверу на наявність нових повідомлень, для цього дописуємо в bot.py наступне(рис. 2.5).

```
if __name__ == '__main__':
    executor.start_polling(dp)
```

Рис. 2.5 Додання постійного опитування серверу

Тепер можна протестувати бота(рис. 2.6).

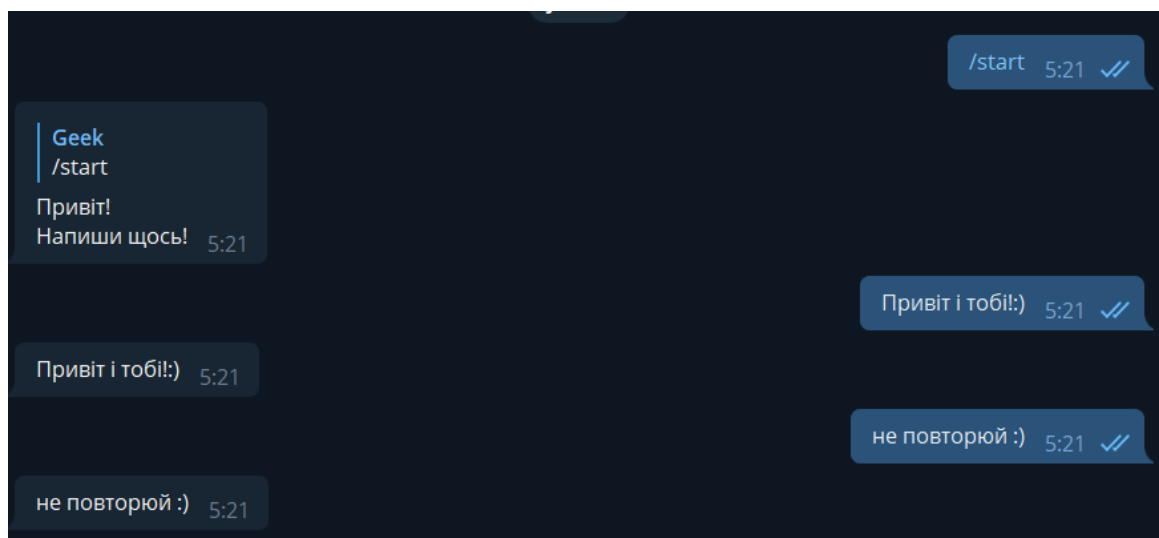


Рис. 2.6 Тестування боту

2.1.3 MongoDB

MongoDB - документо-орієнтована база даних з відкритим вихідним кодом, яка не вимагає опису схеми таблиць. Основні можливості: після того як ми отримали відповідь від сервера – виводимо всі можливі варіанти квитків[13]. В квитку повинні бути:

- підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і розподілу бази даних на вузли;
- документо-орієнтоване зберігання (JSON-подібна схема даних);
- ефективне зберігання великих об'єктів, адміністративний інтерфейс, серверні функції, Map / Reduce і інше;
- використання Javascript в якості мови для формування запитів;

- профілювання запитів;
- широкий набір атомарних операцій над даними (умовний пошук, складна вставка / оновлення тощо);
- різні типи даних (в тому числі підтримка масивів);
- підтримка індексів (B-Tree);
- журналізація операцій, змінюючих дані у базові даних.

Розглянемо кілька властивостей цієї бази даних[13]:

1) Формат даних в MongoDB.

До складу найпопулярніших стандартів обміну даними та їх зберігання входить JSON (JavaScript Object Notation). JSON досить ефективно може описувати складні за структурою дані. В MongoDB досить схожий спосіб зберігання даних на JSON, але формально він тут не використовується. В MongoDB для зберігання даних використовується формат, який має назву BSON, що розшифровується як binary JSON. Цей формат дозволяє швидше працювати з даними, при цьому швидше виконується як пошук, так і сама обробка. Але потрібно зробити примітку, що BSON має недолік в порівнянні з форматом JSON, а саме: дані які зберігаються в форматі BSON займають більше місця ніж дані у форматі JSON.

2) Кросплатформеність.

MongoDB розроблена на мові програмування C++, з цього слідує що перенести її на різні платформи не складе ніяких труднощів. MongoDB можна розгорнути як на Windows, так і на MacOS чи Linux. Також є можливість самому завантажити вихідний код і її скомпілювати, хоча краще використовувати вже спомпільовані бібліотеки які можна завантажити на офіційному сайті.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

3) Колекції.

На відміну від традиційних таблиць в SQL, в MongoDB на зміну їм прийшли колекції. Реляційні бази даних в таблиці можуть зберігати тільки однотипні структуровані об'єкти, а ось колекції здатні зберігати взагалі різноманітні об'єкти з різною структурою та набором властивостей.

4) Реплікація.

Система зберігання даних в MongoDB представляє набір реплік. В цьому наборі є основний вузол, а також може бути набір вторинних вузлів. Усі вторинні вузли зберігають цілісність і автоматично оновлюються разом з оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один з вторинних вузлів стає головним.

5) Простота у використанні.

Те що в MongoDB відсутня жорстка схема даних, і в зв'язку з цим при найменшій зміні концепції в зберіганні даних не потрібно перестворювати цю схему, досить відчутно полегшує роботу з цими базами даних, а також і в подальшому їх масштабуванні.

6) GridFS.

До списку проблем при роботі з будь-якими системами баз даних відноситься проблема збереження даних великого розміру. Як варіант можливо зберігання даних в файлах, використовуючи для цього мови програмування. Деякі СУБД можуть запропонувати спеціальні типи для того щоб зберігати бінарні дані в БД. MongoDB на відміну від реляційних баз даних дозволяє вам зберігати документи з різними даними, але при цьому є обмеження в розмірі документа, 16 Мб. При цьому MongoDB запропонувала рішення – спеціально розроблену технологію яка має назву GridFS, дозволяє обійти це обмеження, і зберігати документи як за розміром більші ніж 16 Мб. Ця система складається з двох колекцій: в одній зберігаються тільки імена

					ІАЛЦ.467100.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

файлів і їх метадані; в іншій колекції зберігаються дані файлів у вигляді невеликих сегментів зазвичай по 256 Кб.

2.4 Середовище розробки JetBrains PyCharm

PyCharm — інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, зручної навігації, інструмент для запуску юніт-тестів. PyCharm розроблена чеською компанією JetBrains в 2010 році. Є версія Community з урізаними можливостями а також версія Edu, яка призначена для навчальних закладів. Обидві версії є безкоштовними. PyCharm працює на всіх популярних операційних системах, таких як Windows, MacOS і Linux.[8] PyCharm надає розумну перевірку коду, швидке виявлення помилок і оперативне виправлення, в купі з автоматичним рефакторингом коду, і багатими можливостями в навігації. Сам редактор коду PyCharm надає першокласну підтримку для Python, JavaScript, CoffeeScript, TypeScript, CSS, популярних мовних і багато чого іншого. Скористайтеся розумним пошуком для переходу до будь-якого класу, файлу, символу, або навіть події IDE і вікна інструментів. Всього один клік, і ви переходите до супер-методу, тестування, використання, реалізації, декларації і це як мінімум! З PyCharm вам відкривається розумне оновлення коду з безпечним видаленням і перейменуванням, методом вилучення, вступної змінної, вбудованої змінної або методом, і іншими видами рефакторинга. Орієнтовані на фреймворк і мову програмування рефакторинг допоможуть виконати будь-яку зміну в рамках цілого проєкту.

Велика колекція інструментів PyCharm включає в себе інтегрований відладчик і запуск тестування:

- профайлер Python;
- вбудований термінал;
- інтеграцію з великими VCS і вбудованими інструментами баз даних;
- можливість віддаленої розробки з віддаленими інтерпретаторами;

					ІАЛЦ.467100.003 ПЗ	Арк.
						31
Зм.	Арк.	№ докум.	Підпис	Дата		

- інтегрований термінал ssh ;
- інтеграція з Docker і Vagrant .

PyCharm також підтримує різні фреймворки для веб розробки, такі як Flask , Django , GoogleAppEngine і web2py, включаючи відладчик шаблонів Django а також інструменти manage.py і appcfg.py, автозаповнення і навігацію. JavaScript також підтримується в PyCharm. PyCharm інтегрується з IPython Notebook, має інтерактивну консоль Python і підтримує як Anaconda, так і безліч інших пакетів, включаючи Matplotlib і NumPy.

Для знайомства з цим середовищем розробки поетапно створимо проєкт в PyCharm.

Створення проєкту:

- 1) Запускаємо PyCharm та натискаємо «Create New Project»(рис. 2.7).

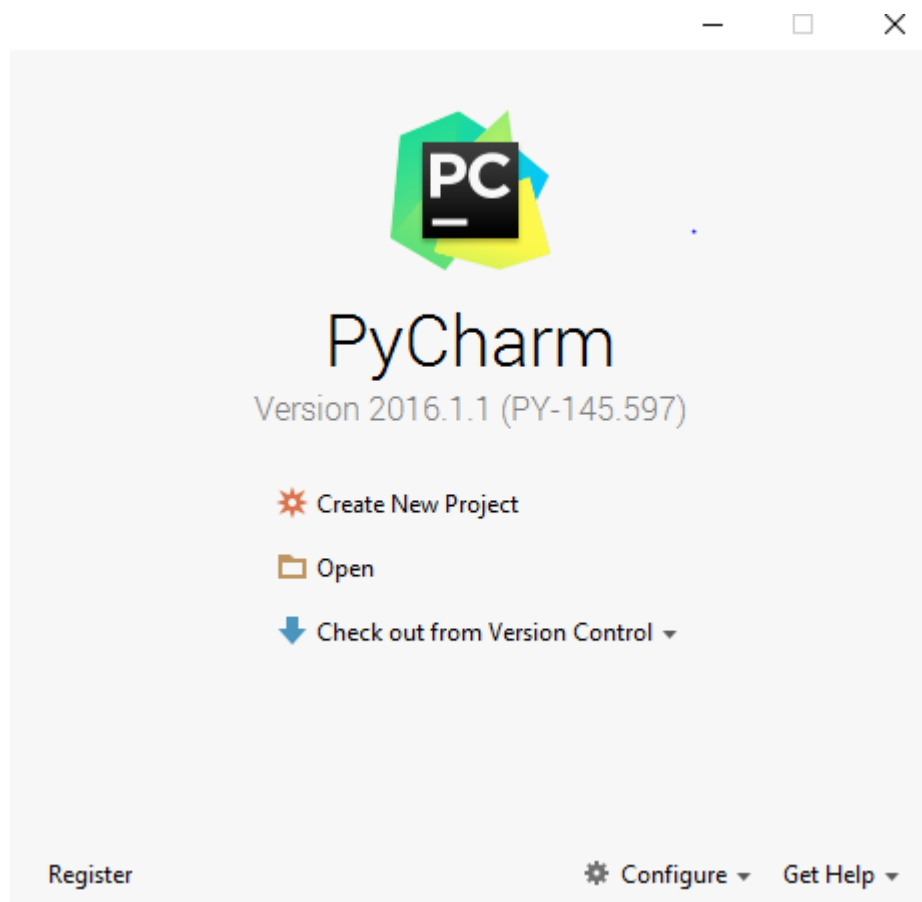


Рис 2.7 Перший етап створення проєкту

2) В наступному вікні вказуємо шлях розташування проєкту та вибрати інтерпретатор»(рис. 2.8).

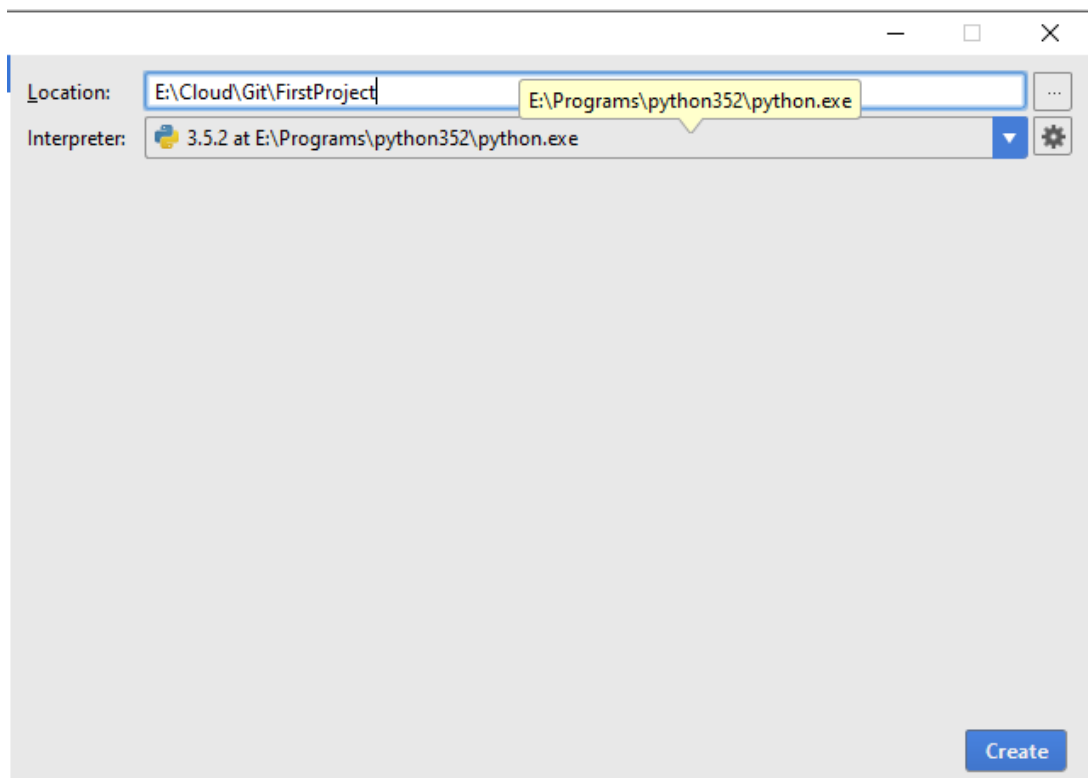


Рис 2.8 Другий етап створення проєкту

3) Далі створюємо файл скрипту»(рис. 2.9).

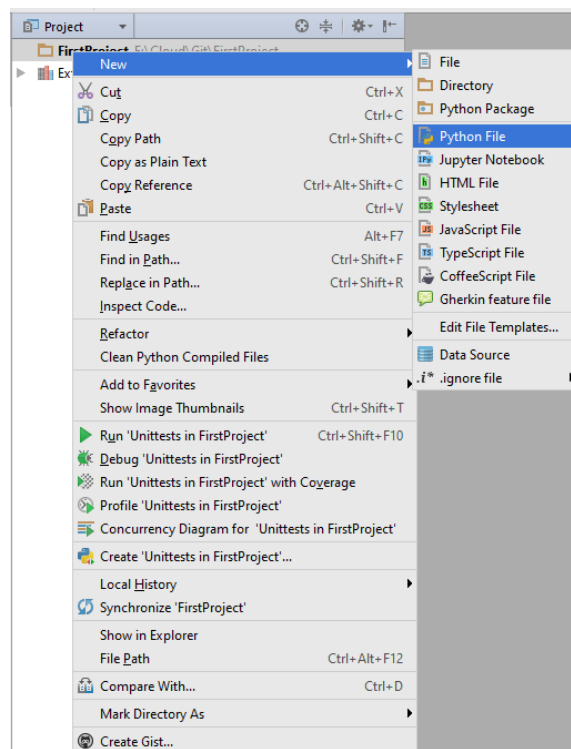


Рис 2.9 Третій етап створення проєкту

Всього за три етапи було створено новий проєкт.

Для кожного проєкту бажано створювати своє віртуальне оточення.

Створення віртуального оточення:

- 1) Заходимо в розділ File-> Settings(рис. 2.10).

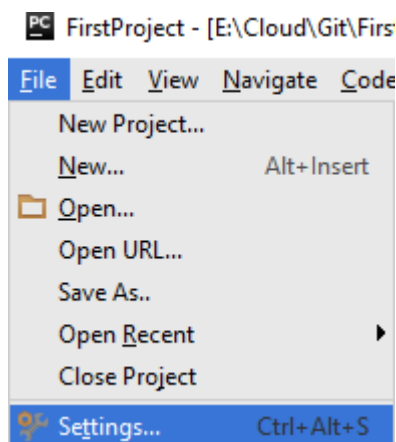


Рис. 2.10 Налаштування проєкту

- 2) Далі Project: Назва вашого проєкту-> Project Interpreter(рис. 2.11).

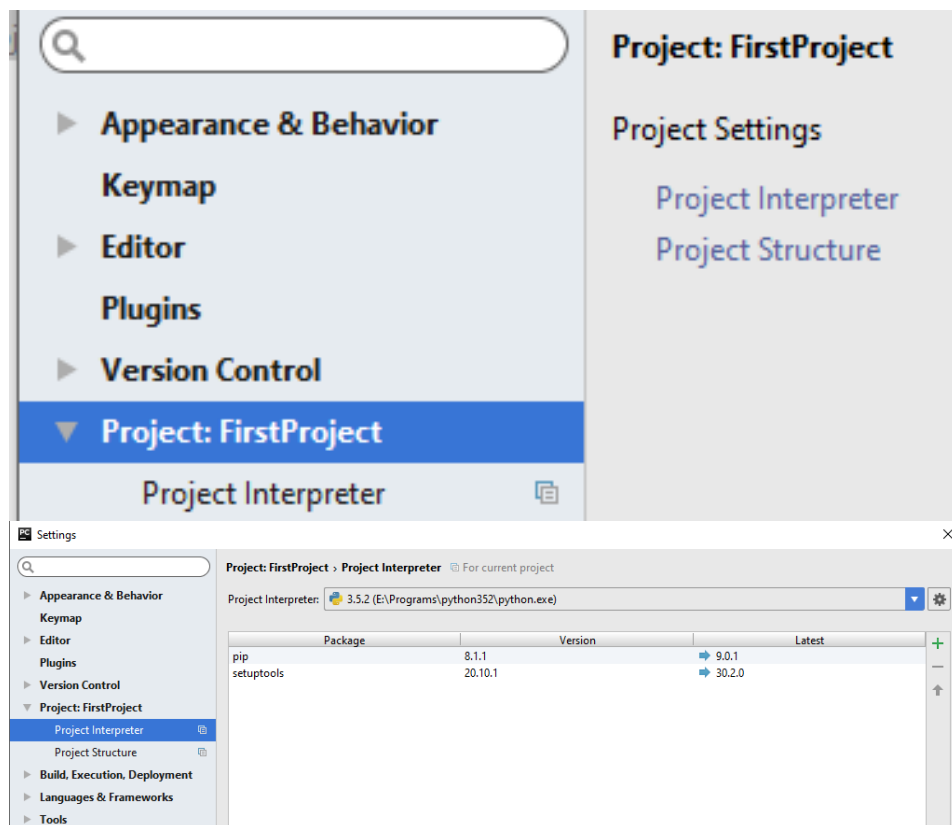


Рис. 2.11 Меню інтерпретатора проєкту

3) Далі Create VirtualEnv»(рис. 2.12).

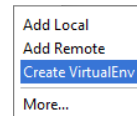


Рис. 2.12 Створення віртуального оточення

4) Задаємо параметри віртуального оточення»(рис. 2.13).

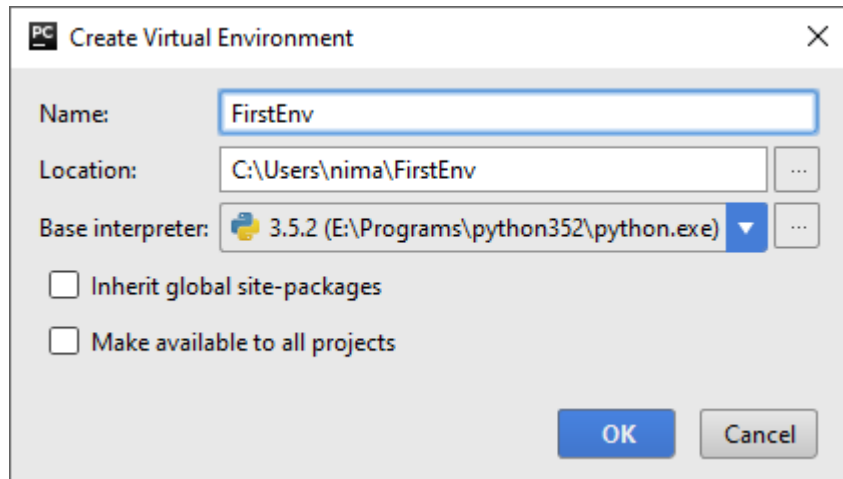


Рис. 2.13 Налаштування віртуального оточення

Все, віртуальне оточення створено і готове до роботи.

ВИСНОВКИ ДО РОЗДІЛУ 2

У цьому розділі були описані вибрані технології та інструменти для розробки чат-боту для проведення тендерів з продажу агрокультур. Описані популярні методи Telegram Bot API. Наведено приклад створення простого чат-боту, використовуючи бібліотеку AIOgram на Python.

Також були наведені етапи створення проєкту та віртуального оточення в середовищі розробки JetBrains PyCharm.

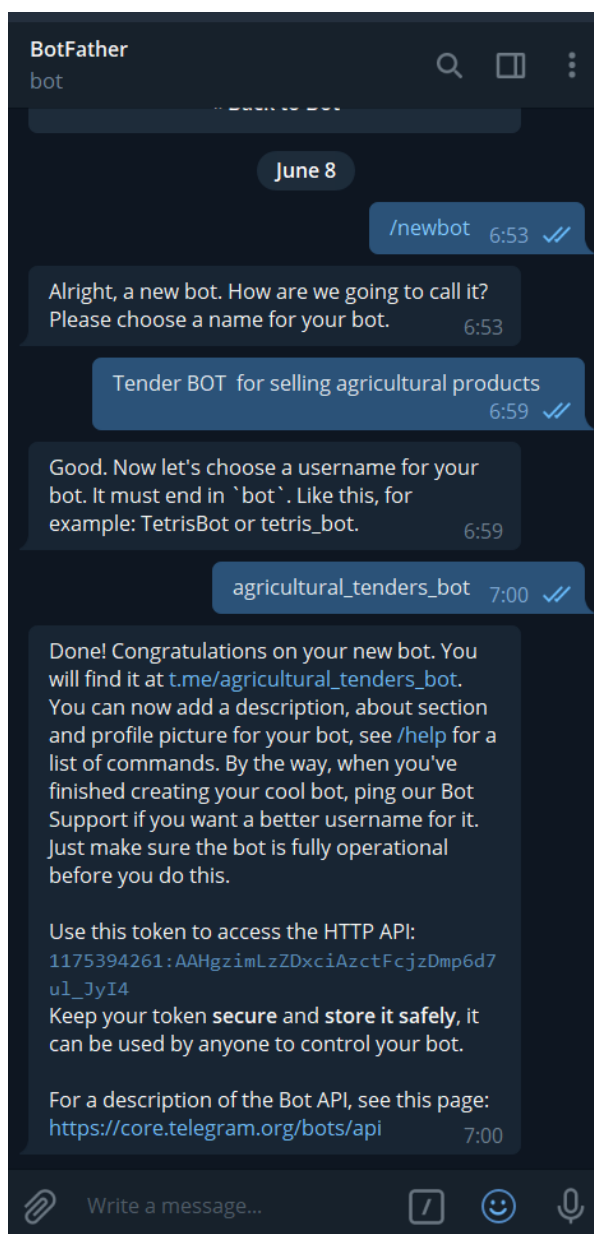
					ІАЛЦ.467100.003 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ЧАТ-БОТУ TELEGRAM

3.1 Отримання токєну

Для того щоб почати розробку бота, потрібно отримати токен. Для цього потрібно зайти в спеціальний бот @Botfather, та за допомогою команди /newbot створити нового бота, далі потрібно ввести назву бота, та юзернейм, який обов'язково повинен закінчуватись на bot.(рис. 3.1). Якщо ви виконали всі умови то бот видає вам токен та URL-посилання для доступу до вашого бота.



					ІАЛЦ.467100.003 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

Рис. 3.1 Створення бота в @Botfather

Крім команди створення бота, він підтримує команди для зміни імені, фото, опису, списку доступних команд телеграм-бота(табл. 3.1).

Табл. 3.1 Команди додаткових налаштувань телеграм-бота

Команда	Дія
/token	Надсилає токен який був отриманий раніше для обраного бота
/setinline	Дозволяє або забороняє звертатись до бота з інших чатів
/setjoingroup	Дозволяє або забороняє додавати бота в групи
/revoke	Видаляє токен для обраного бота

Після створення, налаштування телеграм-бота і отримання токenu можна приступати до написання коду.

3.2 Організація структури файлів бота

Щоб було зручно орієнтуватись по коду, потрібно продумати структуру файлів боту. Підсумкова структура зображена на рисунку 3.2.

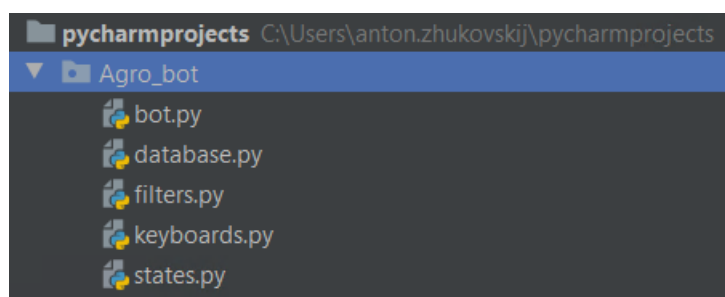


Рис. 3.2 Структура файлів бота

В bot.py будуть описані всі функції та логіка бота. Також створив файл database.py в якому буде винесено код структури БД і її методи. У файлі keyboards.py будуть розміщені всі потрібні клавіатури для бота. У states.py

повинні розміститись стани FSM. В filters.py описаний клас для простішої фільтрації кнопок.

3.3 Хід розробки

Так як структура файлів вже створена, потрібно розробити структури і методи всіх таблиць в БД.

3.3.1 Розробка структури БД

В якості БД для телеграм-бота було обрано MongoDB. Проаналізувавши декілька різних бібліотек на python для взаємодії з цією БД була обрана бібліотека mongoengine. Переваги над іншими бібліотеками: зрозумілі документація та приклади, простота використання. В процесі розробки чат-бота була розроблена схема, яка демонструє взаємодію користувача з БД (рис. 3.3).

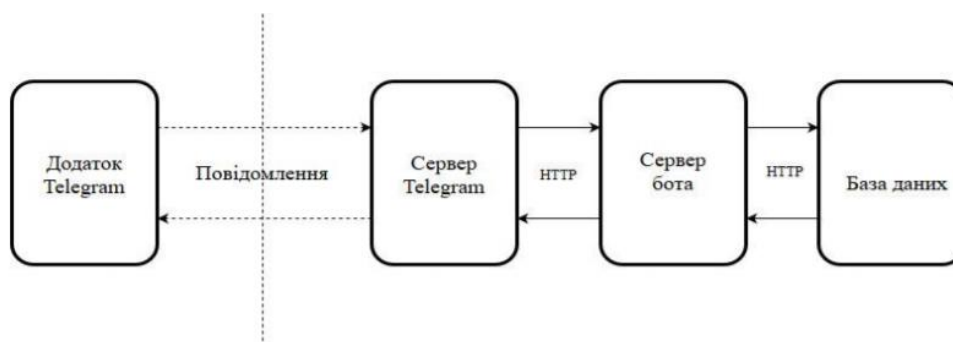


Рис. 3.3 Реалізація взаємодії користувача з БД через телеграм-бота

Також в процесі розробки були створені структури чотирьох документів:

1. Документ User, в якому зберігаються дані про користувача, а саме: ід, юзернейм, ім'я, номер телефону та статус (всі поля зображено на рисунку 3.4).

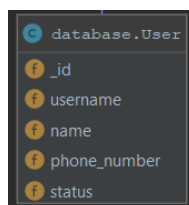


Рис. 3.4 Поля документу User

2. Документ Lot(документ активних лотів), в якому описані поля активного лоту, а саме: поточна ціна, id лоту, порядковий номер лоту, юзернейм автора лоту, назва культури, вага, базис поставки, місяць поставки, валюта, стартова ціна, крок ставки, коментар до лоту, історія ставок, дата створення лоту, хто переглянув лот(всі поля зображено на рисунку 3.5)

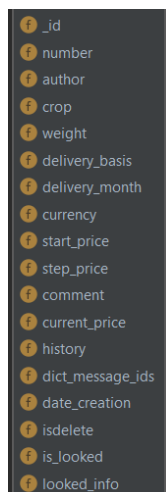


Рис. 3.5 Поля документу Lot

3. Документ Archive(документ закритих лотів), має всі ті ж поля що і у документа Lot (всі поля зображено на рисунку 3.6)

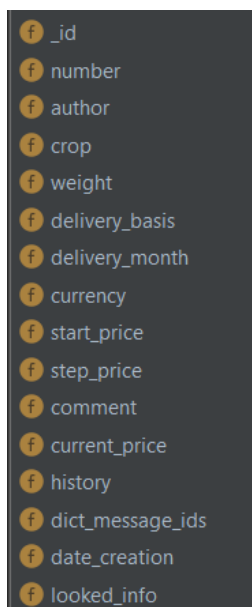


Рис. 3.6 Поля документу Archive

4. Документ Settings(документ основних налаштувань), включає в собі: останній номер лоту, номери телефонів адміністраторів, доступні

культури, доступні бази, номери телефонів кому закритий доступ, номери телефонів кому відкритий доступ (всі поля зображено на рисунку 3.7)



Рис. 3.7 Поля документу Settings

Разом із структурою були розроблені методи для документу Lot(рис. 3.8).

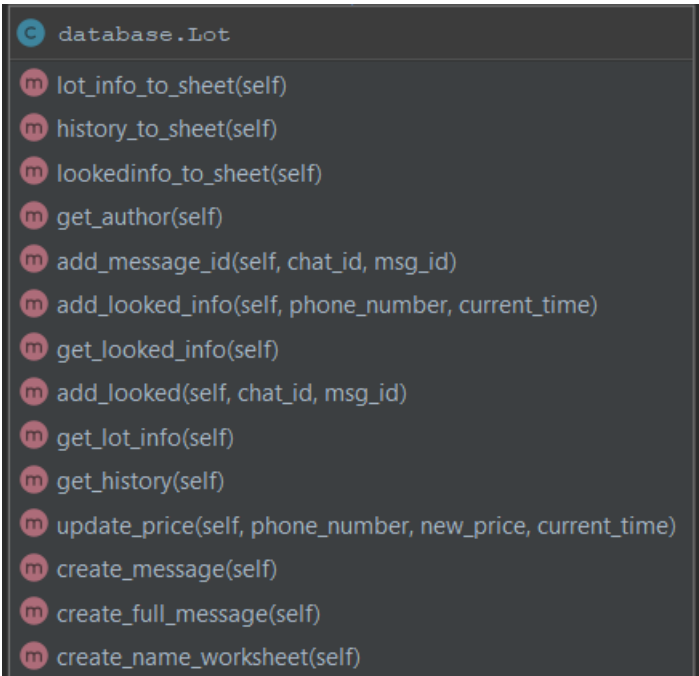


Рис. 3.8 Методи документу Lot

Опис методів наведено в таблиці 3.2.

Табл. 3.2 Опис методів документу Lot

Назва методу	Опис методу
lot_info_to_sheet	Повертає список з інформацією по лоту, для додавання до таблиці
history_to_sheet	Повертає список з історією ставок, для додавання до таблиці
lookedinfo_to_sheet	Повертає список користувачів, як переглянули лот, для додавання до таблиці
get_author	Повертає юзернейм автора лоту

Табл. 3.2(продовження) Опис методів документу Lot

add_message_id	Додає id відправленого повідомлення
add_looked_info	Додає інформацію про користувачів, які переглянули лот
get_looked_info	Повертає інформацію про користувачів які переглянули лот
add_looked	Додає id користувачів які переглянули лот
get_lot_info	Повертає інформацію по лоту
get_history	Повертає історію ставок лоту
update_price	Оновлює поточну ціну лоту
create_message	Повертає готове повідомлення про створення нового лоту, для відправки користувачу
create_full_message	Повертає готове повідомлення з повною інформацією по лоту, для відправки користувачу
create_name_worksheet	Генерує назву для листа в таблиці

3.3.2 Індивідуальні налаштування бота

Так як такі дані як номери телефонів адміністраторів, доступні культури, доступні базиси, білий та чорний список номерів постійно потребують оновлення, для зручності було вирішено перенести ці налаштування в таблицю GS. Такий варіант ідеально підходить, адже не потрібно бути програмістом щоб змінити налаштування, та й на сьогоднішній час більшість людей хоч раз працювали з таблицями Excel або ж GS. Це максимально спрощує процес зміни цих налаштувань. Приклад структури таблиці з налаштуваннями показано на рисунку 3.9.


<div>  tender_telegram_bot ☆ 🔍 ☁ </div> <div> Файл Редагувати Вигляд Вставити Формат Дані Інструменти Допс </div>					
<div> ↶ ↷ 🖨 🔍 100% р. % .0 .00 123 Calibri 11 B </div>					
<div> fx </div>					
	A	B	C	D	E
1	тел. адмінів	культури	базиси	білий список	чорний список
2	380683358425	Corn	Mykolaiv Port	380683358425	
3	380674409230	Soya Beans	Odessa Port	380952747076	
4	380673066398	Rapeseed	Pivdennyi Port	380503348870	
5	380952747076	Rye	Chornomorsk Port	380503574126	
6	380672262604	Sunflower	Illichivsk Sea Port	380677363144	
7		Barley		380673345816	
8		Wheat 2 nd class		380674682215	
9		Wheat 3 d class		380504122936	
10		Wheat 4 th class		380633153787	
11				380977286796	
12				380953659954	
13				380504147730	

Рис. 3.9 Структура таблиці налаштувань

Після того як адміністратор змінив дані в таблиці, йому потрібно перейти в бота та натиснути кнопку «Update from google sheet», після чого всі зміни автоматично перенесуться з таблиці в БД. Взаємодія із GS була реалізована з допомогою бібліотеки gspread. Для того щоб почати працювати з таблицями через API, потрібно пройти автентифікацію та авторизувати свою програму. Для його отримання потрібно:

1. Увімкнути доступ до API для проєкту.
2. Перейти до «APIs & Services > Credentials» і вибрати «Create credentials > Service account key».
3. Повністю заповнити форму.
4. Натиснути «Create key».
5. Вибрати «JSON» та натиснути «Create».

6. Завантажити файл JSON в папку з проєктом, та в основний файл скрипту додати код авторизації(рис. 3.10).

```
from oauth2client.service_account import ServiceAccountCredentials

scope = ['https://spreadsheets.google.com/feeds']
credentials = ServiceAccountCredentials.from_json_keyfile_name(
    r'Розташування JSON файлу', scope)
gc = gspread.authorize(credentials)
```

Рис. 3.10 Код авторизації для доступу до GS API

Після цього підключитись до потрібної таблиці за допомогою методу `gc.open()`, і можна починати працювати з таблицею.

3.3.3 Реєстрація в боті

Так як нам потрібно контролювати доступ до боту, була додана реєстрація. Після надсилання боту команди `/start`, бот перевіряє в БД чи є ця людина в базі, якщо немає пропонує зареєструватись, відправивши свій номер, якщо ж користувач уже зареєстрований, відкриває доступ в залежності від ролі користувача(адміністратор або клієнт).

Для того щоб не можна було обійти систему, відправивши чужий номер телефону, відправка телефону була реалізована кнопкою(рис 3.10), і також додана умова порівняння `id` номеру та `id` користувача(вони повинні бути однакові, таку інформацію надає сам telegram). Завдяки цьому зареєструватись під номером телефону до якого немає доступу неможливо.

```
if message.from_user.id == message.contact.user_id:
    phone_number = message.contact.phone_number.replace('+', '')
    list_admins = database.Settings.objects(_id='1').first().admin_phone_numbers
    if phone_number in list_admins:
        status = 'admin'
    else:
        status = 'client'
    user = database.User(_id=str(message.chat.id), username=message.from_user.username,
                        name=message.from_user.full_name,
                        phone_number=phone_number, status=status).save()
    await states.Menu.main_menu.set()
    msg = await bot.send_message(message.chat.id, 'done ', reply_markup=keyboards.hide_markup)
    await bot.delete_message(msgs.chat.id, msgs.message_id)
    if user.status == 'client':
        lots = database.Lot.objects()
        for lot in lots:
            msg = await bot.send_message(message.chat.id, text=lot.create_message(),
                                        reply_markup=keyboards.send_hidden_lot(lot._id))
            lot.add_message_id(msg.chat.id, msg.message_id)
            lot.save()
    elif user.status == 'admin':
        await states.Menu.admin_menu.set()
        await bot.send_message(message.chat.id, '👁', reply_markup=keyboards.admin_keyboard())
    else:
        await bot.send_message(message.chat.id, 'Do not cheat', reply_markup=keyboards.hide_markup)
        await bot.send_message(message.chat.id, 'Send your number👁📱',
                                reply_markup=keyboards.phone_number_keyboard())
```

Рис. 3.10 Реалізація захисту від відправки чужого номеру телефону

					ІАЛЦ.467100.003 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

3.3.4 Опис інтерфейсу адмін-панелі

Інтерфейс адмін-панелі налічує чотири кнопки(рис. 3.11):

- Create a lot – створення лоту
- Active lots – перегляд активних лотів
- Closed lots – перегляд закритих лотів
- Update from google sheet – оновити дані з GS

При натисканні кнопки «Active lots» виводяться всі активні лоти, з можливістю по кожному лоту: поставити ставку, подивитись історію ставок, подивитись хто переглядав лот, і закрити лот(рис. 3.11).

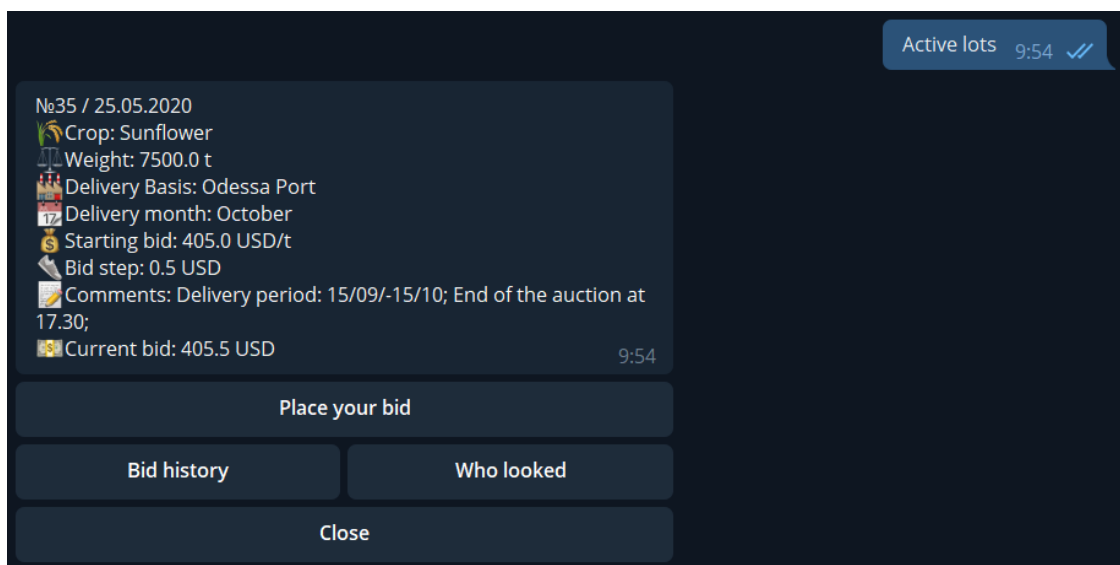


Рис 3.11 Інтерфейс кнопки «Active lots»

					ІАЛЦ.467100.003 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

При натисканні кнопки «Closed lots» виводяться всі закриті лоти, з можливістю по кожному лоту: подивитись історію ставок, подивитись хто переглядав лот а також повторно відкрити лот(рис. 3.12).

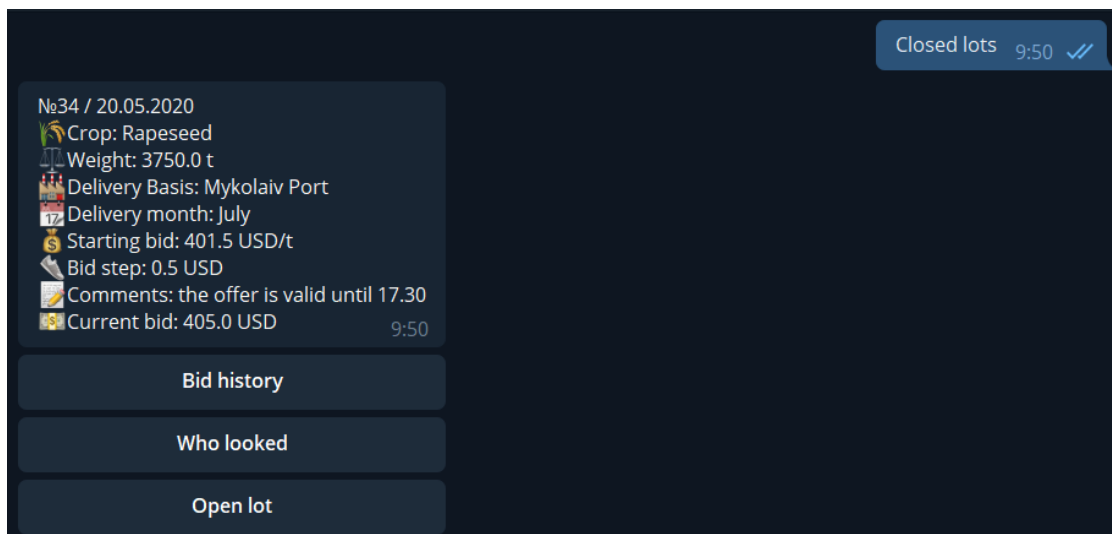


Рис 3.12 Інтерфейс кнопки «Closed lots»

При натисканні кнопки «Create a lot» адміністратор переходить до меню створення нового лоту. Бот послідовно запитує в користувача всю потрібну інформацію для формування нового лоту. Процес створення лоту має таку послідовність:

1. Вибір культури.

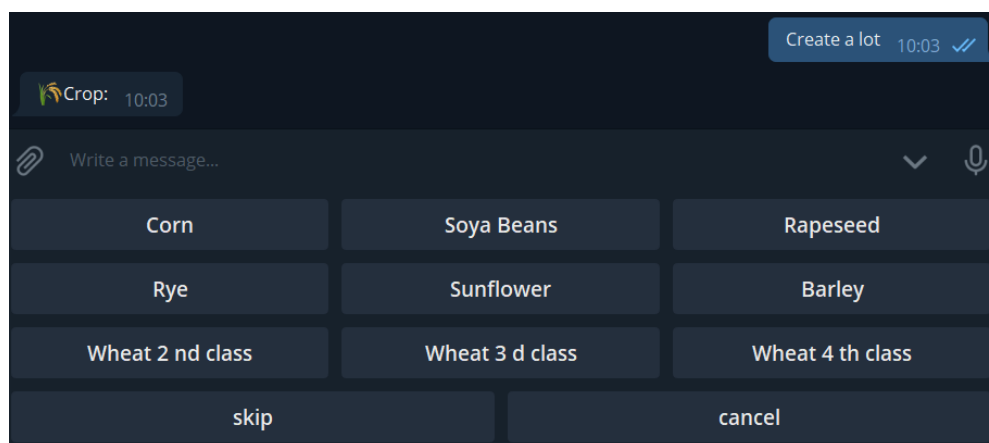


Рис. 3.13 Меню вибору культури

2. Вага культури, т.

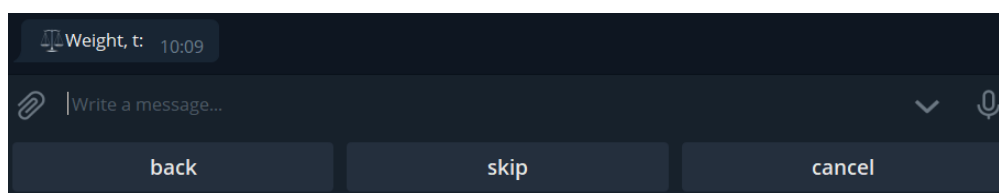


Рис. 3.14 Введення ваги культури

					ІАЛЦ.467100.003 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Базис поставки.

Рис. 3.15 Вибір базису поставки

4. Місяць поставки.

Рис. 3.16 Вибір місяця поставки

5. Вибір валюти.

Рис. 3.17 Вибір валюти

6. Початкова ціна.

Рис. 3.19 Введення початкової ціни

7. Мінімальний крок ставки.

					ІАЛЦ.467100.003 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

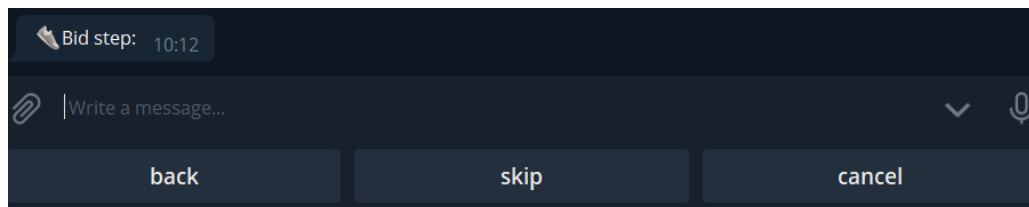


Рис. 3.19 Введення мінімального кроку ставки

8. Коментар до лоту

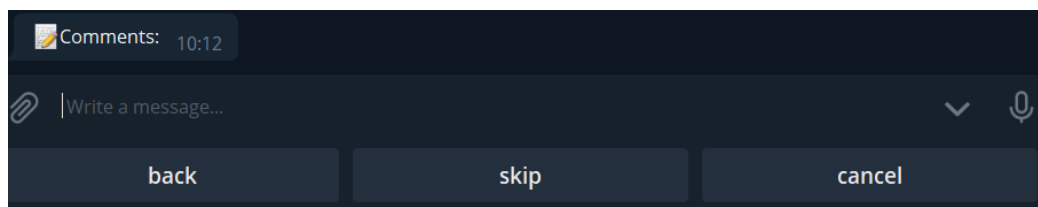


Рис. 3.20 Додання коментарю до лоту

Після цього якщо всі дані введені коректно, всім користувачам бота приходить пуш про появу нового лоту, з можливістю зробити ставку.

3.3.5 Опис інтерфейсу клієнта

Взагалі система діє по типу підписки - після реєстрації користувачу одразу приходять всі активні лоти але з обмеженою інформацією та кнопкою «Show more», це зроблено для того щоб моніторити хто переглянув лот. Після натискання кнопки «Show more» користувач отримує повну інформацію по лоту і одразу може робити ставки. Після закриття лоту, повідомлення з цим лотом видаляються у всіх користувачів.

3.3.6 Опис алгоритму проведення торгів

Взагалі для проведення тендерів потрібна система яка за своїм функціоналом нагадує аукціон. Користувачі повинні мати можливість робити ставки, а бот повинен максимально швидко реагувати на зміну ціни та у режимі реального часу змінювати поточну ціну у всіх користувачів. Вивчивши всі можливості та обмеження telegram bot api, вийшло розробити алгоритм, який задовільняє всі потреби.

Алгоритм працює таким чином – при створенні лоту, бот автоматично розсилає всім користувачам повідомлення з короткою інформацією про лот та кнопкою «Show more» (ід кожного повідомлення записується в БД, для

					ІАЛЦ.467100.003 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

подальшої можливості їх редагування), коли користувач її натиснув, він додається в список переглянувших цей лот, і це повідомлення редагується, відкриваючи повну інформацію про лот для користувача, а також додає можливість зробити ставку. З цього часу при будь-яких змінах бот проходить циклом по всім повідомленням(які стосуються цього лоту) і редагують їх підставляючи актуальну інформацію. Тобто коли будь який користувач додає ставку, бот перевіряє коректність ставки і якщо все окей, моментально редагує повідомлення у всіх користувачів.

3.3.8 Функція експорту закритих тендерів в Google Sheets

Всі закриті лоти зберігаються в БД, але для можливості зручного перегляду людям без навичок програмування, а також для зручного надання доступу певним користувачам, було прийнято рішення дублювати закриті лоти в spreadsheet.

Була розроблена функція, яка викликається при закритті лоту, і експортує всю інформацію по лоту в аркуш spreadsheet. Код даної функції зображений на рисунку 3.21.

```
def add_lot_to_sheet_archive(lot_id):
    gc.login()
    logging.info(f'start add to sheet archive lot[{lot_id}]')
    try:
        lot = database.Archive.objects(_id=lot_id).first()
        ws = archive.add_worksheet(title=f"{lot.create_name_worksheet()}", rows="1000", cols="20")
        cell_list = ws.range('A1:K2')
        val_list = lot.lot_info_to_sheet()
        i = 0
        for cell in cell_list:
            cell.value = str(val_list[i])
            i += 1
        ws.update_cells(cell_list)
        n = len(lot.history)
        n = n + 3
        cell_list = ws.range(f'A3:D{n}')
        val_list = lot.history_to_sheet()
        i = 0
        for cell in cell_list:
            cell.value = str(val_list[i])
            i += 1
        ws.update_cells(cell_list)
        n1 = len(lot.looked_info)
        cell_list = ws.range(f'A{n + 1}:C{n + n1 + 1}')
        val_list = lot.lookedinfo_to_sheet()
        i = 0
        for cell in cell_list:
            cell.value = str(val_list[i])
            i += 1
        ws.update_cells(cell_list)
        logging.info(f'end add to sheet archive lot[{lot_id}]')
    except Exception as e:
        logging.error(f'{e}')
```

Рис. 3.21 Функція експорту закритих лотів в spreadsheet

					ІАЛЦ.467100.003 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

Функція приймає один аргумент – id лоту, і вже сама витягує всю інформацію з БД, і завантажує в таблицю, створюючи новий аркуш.

Приклад зображений на рисунку 3.22.

	A	B	C	D	E	F	G	H	I	J	K
1	Культура	Вага	Базис доставки	Місяць поставки	Валюта	Стартова ціна	Крок	Поточна ціна	Коментар	Автор лоту	Дата створення
2	Rapeseed	3750.0	Mykolaiv Port	July	USD	401.5	0.5	405.0	the offer is valid	657699837	2020-05-20 16:38:50.680000
3	Телефон	Ціна	Час	Дата							
4	380503538690	404.0	16:43:11	20.05.2020							
5	380503210169	405.0	17:29:24	20.05.2020							
6	Телефон	Час	Дата								
7	380673066398	16:38:55	20.05.2020								
8	380683358425	16:38:56	20.05.2020								
9	380503554688	16:39:02	20.05.2020								
10	380674663701	16:39:03	20.05.2020								
11	380952747076	16:39:03	20.05.2020								
12	380674409230	16:39:04	20.05.2020								
13	380677363144	16:39:06	20.05.2020								
14	380633153787	16:40:47	20.05.2020								
15	380977286796	16:41:25	20.05.2020								
16	380503538690	16:42:05	20.05.2020								
17	380503210169	16:42:41	20.05.2020								
18	380500132308	16:43:51	20.05.2020								
19	380503574126	16:54:34	20.05.2020								
20	380674079889	20:23:57	20.05.2020								
21											
22											
23											
24											
25											
26											

Рис. 3.22 Приклад експорту закритого лоту

3.5 Розміщення та запуск проєкту на сервері

3.5.1 Встановлення Python

Для роботи нашого проєкту нам потрібен Python версії 3.7 та вище. Для цього потрібно перейти на офіційний сайт, вибрати вкладку «Downloads» і натиснути «Download Python» (рис. 3.22).



Рис. 3.22 Завантаження останньої актуальної версії Python

Після запуску файлу інсталяції відкриється вікно з налаштуваннями установки (рис. 3.23), ставимо галочку біля опції «Add Python 3.7 to Path» (це дозволить нам відкривати консоль, не вказуючи повний шлях в командному рядку), і жмемо «Customize installation».

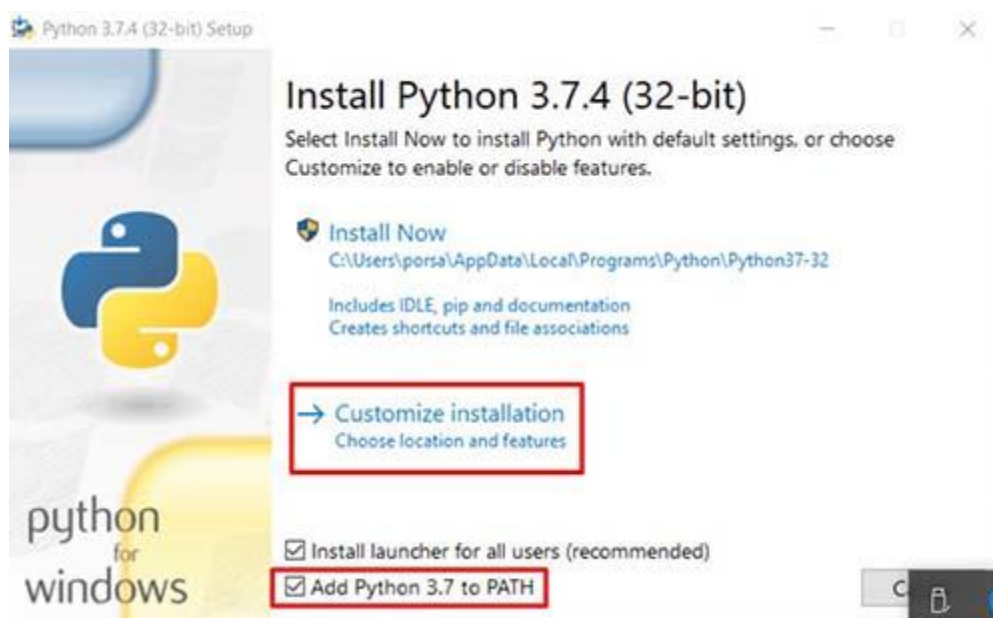


Рис. 3.23 Вибір способу встановлення Python

На наступному вікні потрібно відмітити всі галочки (рис. 3.24), додатково встановлюються документація, пакетний менеджер та IDLE. Якщо всі галочки помічені, тиснемо «Next».

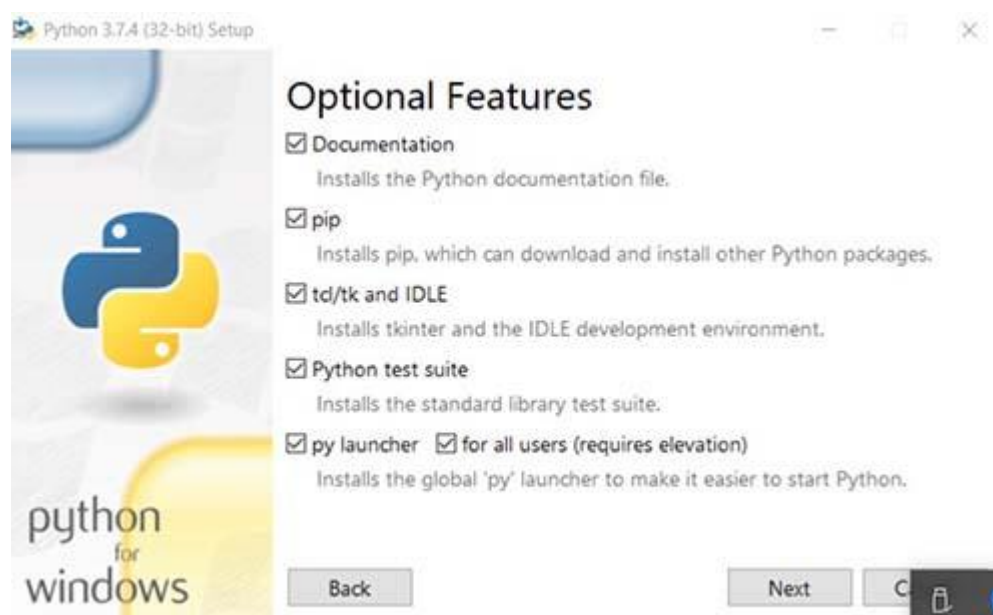


Рис. 3.24 Вибір опцій при встановленні Python

На наступному вікні залиште відмічені галочки за замовчуванням та вкажіть шлях, куди буде встановлений Python (рис.3.25) ,і натискайте «Install».

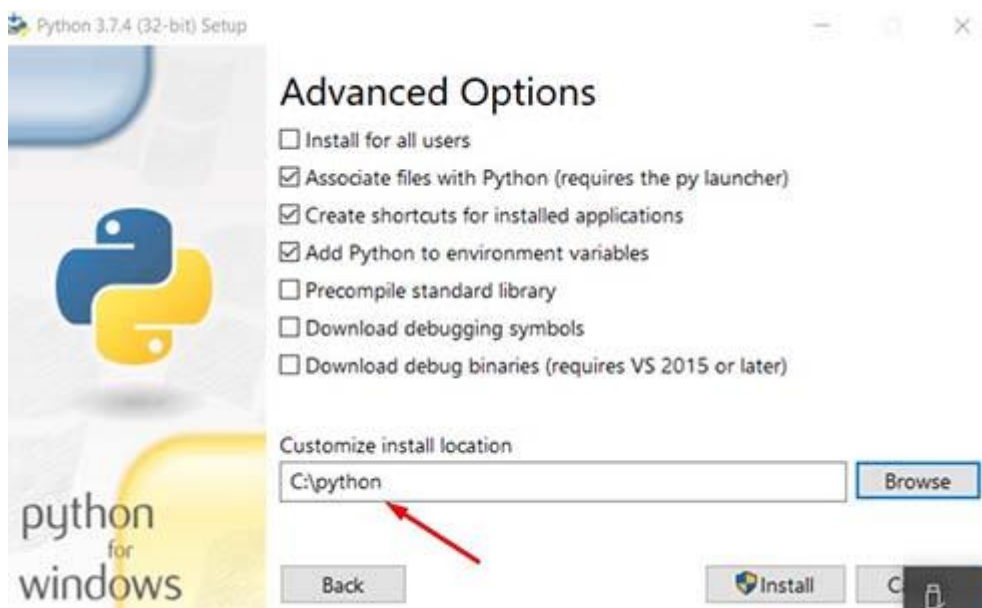


Рис. 3.25 Вибір директорії для встановлення Python

Якщо все зроблено правильно, з'явиться наступне вікно(рис.3.26).

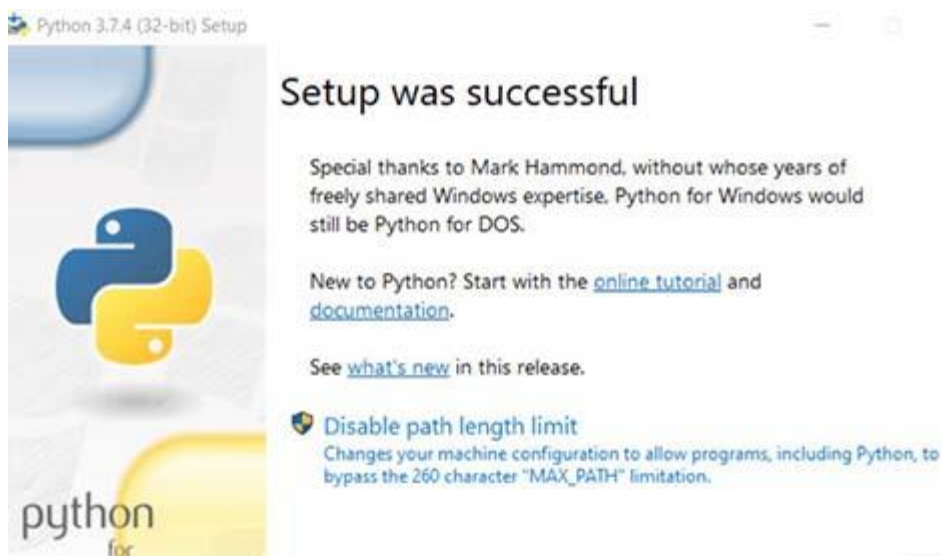


Рис. 3.26 Вікно завершення встановлення Python

3.5.2 Встановлення необхідних бібліотек

Після того, як Python успішно встановлено, потрібно встановити всі потрібні бібліотеки. Для цього на локальному комп'ютері в директорії проєкту генеруємо requirements.txt за допомогою команди `pip freeze > requirements.txt`, копіюємо весь проєкт на сервер і встановлюємо всі бібліотеки за допомогою

					ІАЛЦ.467100.003 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

команди `pip install -r requirements.txt` (команди виконувати в командному рядку знаходячись у директорії проєкту).

3.5.3 Реалізація автоматичного запуску бота при перезавантаженні сервера

Інколи сервера можуть самотійно перезавантажуватись, і бот просто перестане відповідати на запити. Щоб такого не відбувалось скористаємось стандартною утилітою Windows Task Scheduler.

Для початку створимо *.bat файл, в середині якого потрібно записати дві команди – перехід в директорію проєкту, та запуск файла bot.py. Назвемо його для прикладу start_agrobot.bat.

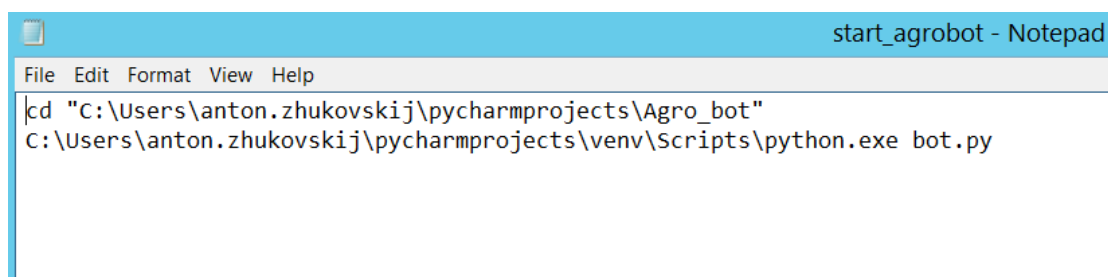


Рис. 3.27 Приклад вмісту *.bat файлу

Тепер запусимо Task Scheduler та натиснемо «Create Basic Task Wizard». Після цього відкриється вікно де потрібно ввести ім'я задачі та опис (рис. 3.28) і нажимаємо «Next». В наступному вікні вибираємо пункт «When the computer starts» (рис. 3.29) і нажимаємо «Next».

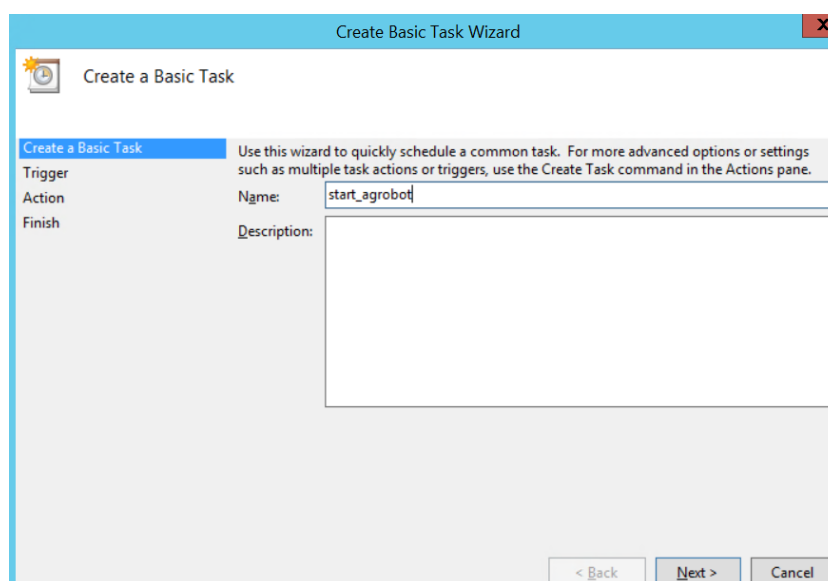


Рис. 3.28 Створення імені та опису задачі

Зм.	Арк.	№ докум.	Підпис	Дата

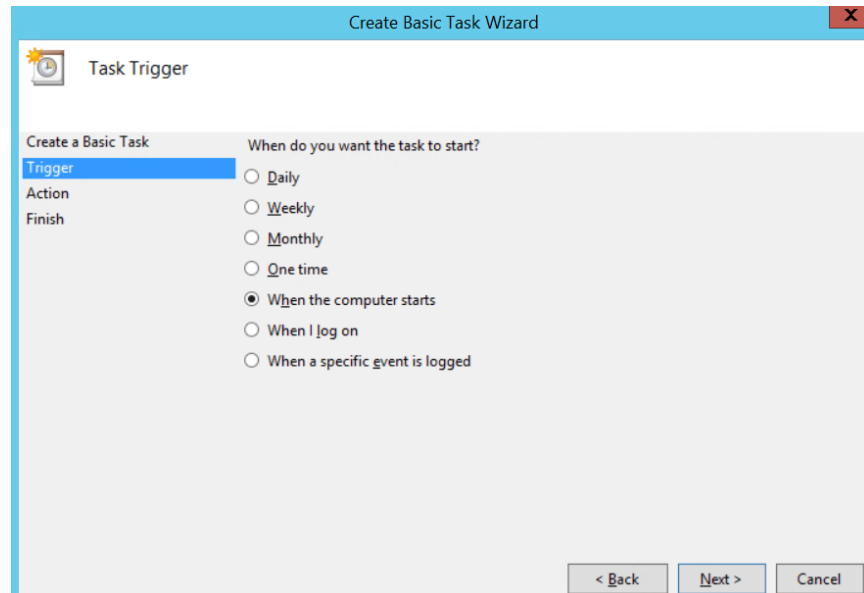


Рис. 3.29 Вибір умови для старту задачі

Тепер потрібно вибрати тип дії «Start a program». В наступному вікні додаємо шлях до нашого *.bat файлу і нажимаємо «Next»(рис. 3.29).

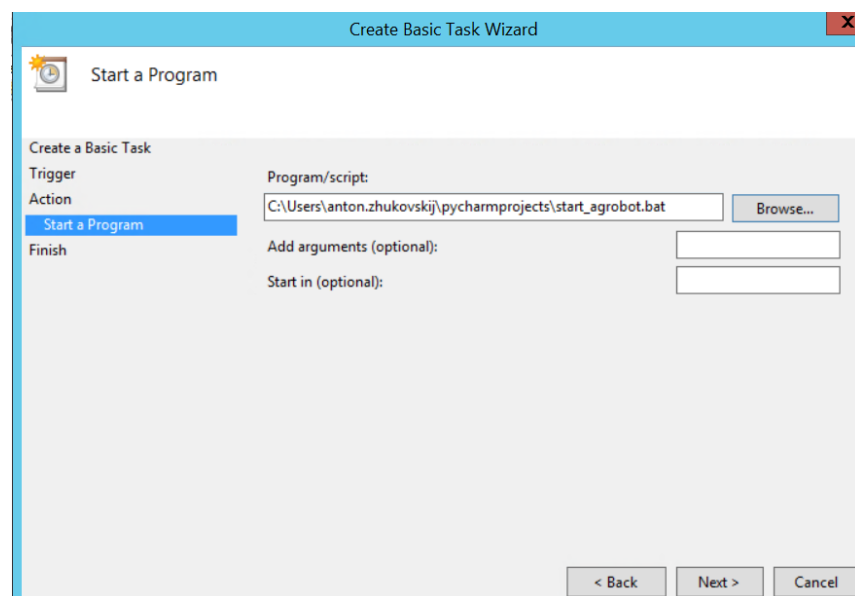


Рис. 3.29 Вибір потрібного скрипту для запуску

Останній раз перевіряємо чи все правильно введено(рис. 3. 30) і натискаємо «Finish».

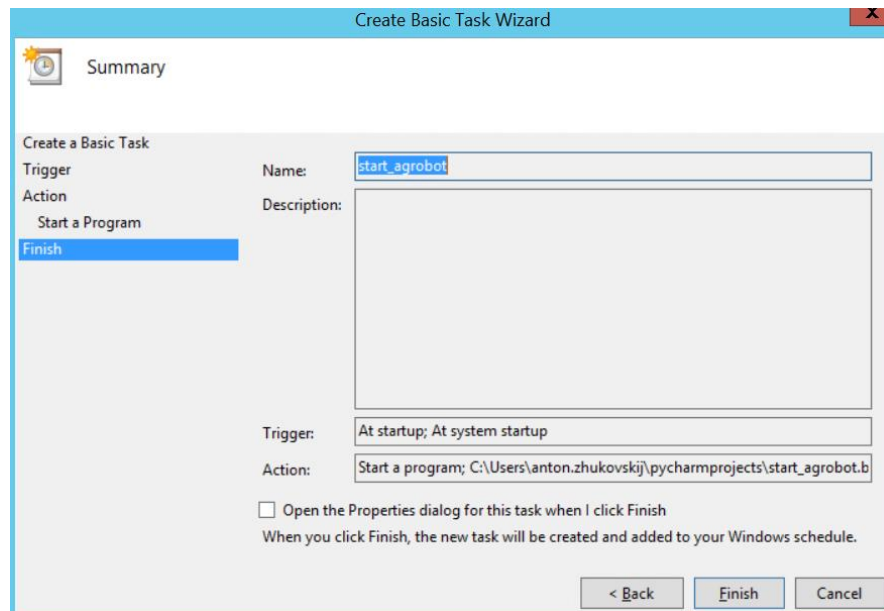


Рис. 3.30 Перевірка введених даних

Готово, тепер при перезавантаженні серверу скрипт автоматично запустить бота.

ВИСНОВОК ДО РОЗДІЛУ 3

У даному розділі було описано хід розробки телеграм боту, а саме розробки структури БД, інтеграція з GS, реалізації реєстрації, інтерфейсу адмін-панелі, інтерфейсу клієнта, основного алгоритму проведення торгів. \ Реалізована зручна структура проєкту. А також описані етапи розміщення та запуску проєкту на сервері.

					ІАЛЦ.467100.003 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Розробка даного дипломного проєкту була присвячена створенню телеграм-боту для проведення тендерів з продажу агрокультур. Програмний продукт написаний на мові програмування Python, використовуючи бібліотеки AIOgram, gspread, mongoengine, asyncio.

У ході виконання проєкту були розглянуті світові інформаційні тенденції та проаналізовані самі актуальні месенджери.

Також був проведений аналіз предметної області даної роботи та було визначено основні вимоги для розробки проєкту.

Беручи до уваги вимоги, які задані вище, було проведено опис та аналіз використаних технологій для реалізації системи.

Реалізація проєкту відбувалась враховуючи функціонал та вимоги до розробки, які зазначені в технічному завданні, а саме реалізації системи проведення тендерів з продажу агрокультур.

Основним недоліком даної реалізації є те, що Telegram дозволяє редагувати повідомлення тільки 2 дні після їх відправлення. Так як основний алгоритм торгів використовує редагування для миттєвого відображення змін, то один лот повинен бути активним не більше 2х днів, тому як користувачам просто перестануть приходити зміни.

					ІАЛЦ.467100.003 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Сухас У. Oracle Intelligent Bots: Чат-боты с искусственным интеллектом [Електронний ресурс] / У. Сухас. – Oracle, 2017. – Режим доступу до ресурсу: <https://www.oracle.com/a/ocom/docs/chatbots.pdf>.
2. A curated directory of chat bot resources & tools [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.botsfloor.com/botstash/products/?category=Development%20Platforms>.
3. Beaver L. The Chatbots Explainer [Електронний ресурс] / L. Beaver. – BI Intelligence Copyright, 2016. – 23 p. – Режим доступу до ресурсу: <https://www.businessinsider.com/intelligence/chatbots-explainer>.
4. Chatbot Report 2018: Current landscape of how people create chatbots and how users expect to interact with them [Електронний ресурс]. – Режим доступу до ресурсу: <https://elearningindustry.com/chatbots-for-learning-support-10-reasons>.
5. Goncalves P. 10 graphs that show why your business should be available through messaging apps [Електронний ресурс] / P. Goncalves. – Режим доступу до ресурсу: <https://medium.com/hijiffy/10-graphs-that-show-the-immense-power-of-messagingapps-4a41385b24d6>.
6. Kumar C. 9 AI Platform to Help you in Creating Facebook Chatbot [Електронний ресурс] / C. Kumar. – Режим доступу до ресурсу: <https://geekflare.com/create-facebook-chatbot/>.
7. On Their Tenth Anniversary, Mobile Apps Start Eating Their Own [Електронний ресурс]. – Режим доступу до ресурсу: <https://flurrymobile.tumblr.com/post/155761509355/on-their-tenth-anniversary-mobile-apps-start>.
8. Путівник мовою програмування Python [Електронний ресурс] – Режим доступу до ресурсу: <https://pythonguide.rozh2sch.org.ua/>.
9. Справочник по Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://tlgrm.ru/docs/bots/api>.

					<i>ІАЛЦ.467100.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

10. Python лідер з вивчення мов програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://www.armedsoft.com/ua/blog/python-lider-z-vyvchennya-mov-programuvannnya>.
11. Переваги мови Python [Електронний ресурс] – Режим доступу до ресурсу: <https://disted.edu.vn.ua/courses/learn/7649>.
12. Telegram Bot API [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api>.
13. Онлайн-руководство по MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/nosql/mongodb/>.

					ІАЛЦ.467100.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

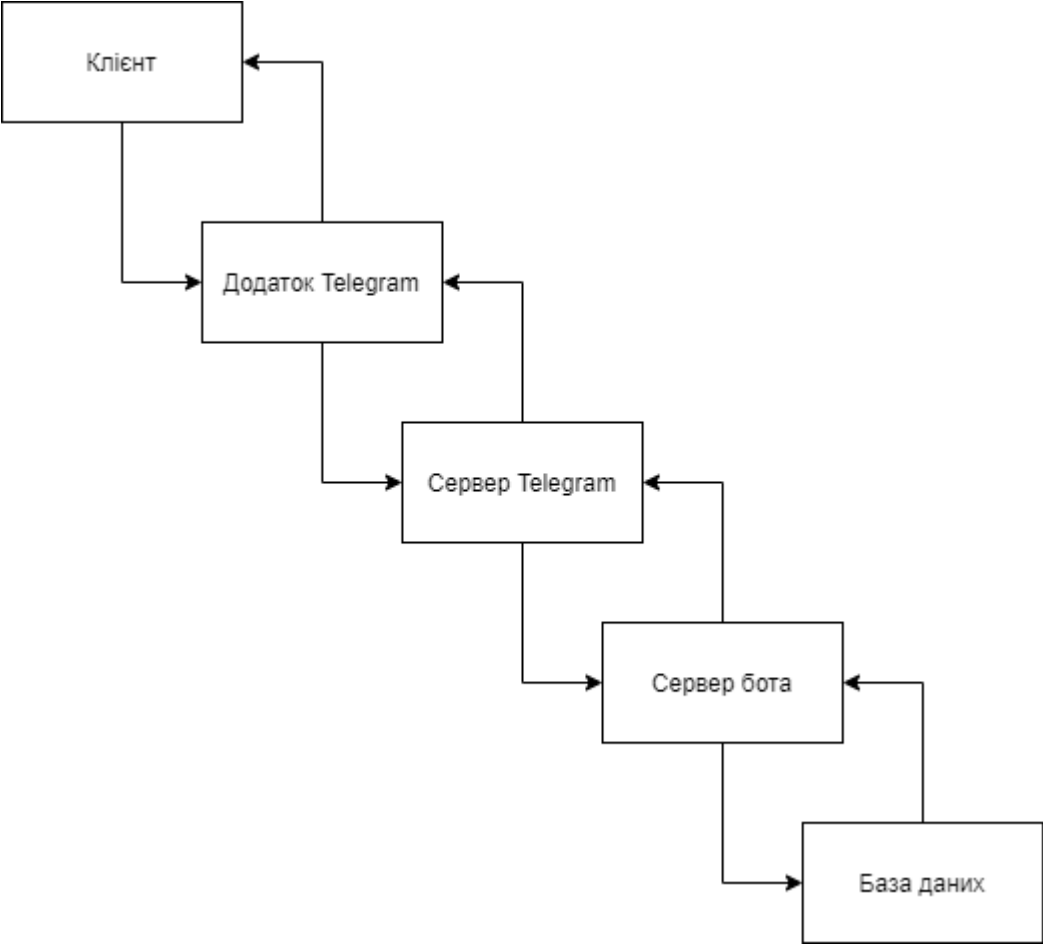
ДОДАТОК 1

Бот для проведення тендерів з продажу агрокультур

Схема структурна
ІАЛЦ.467100.004 Д1

Аркушів 1

Київ — 2020 р.



					ІАЛЦ.467100.004 Д1					
Зм.	Арк.	№ докум.	Підпис	Дата	Бот для проведення тендерів з продажу агрокультур Схема структурна			Літ.	Аркуш	Аркушів
Розробив		Жуковський А.Л.								
Перевішив		Габінет А.В.								
Реценз.								НТУУ «КПІ», ФІОТ, ІО-64		
Н. Контр.		Сімоненко В.П.								
Затв.		Стіренко С.Г.								

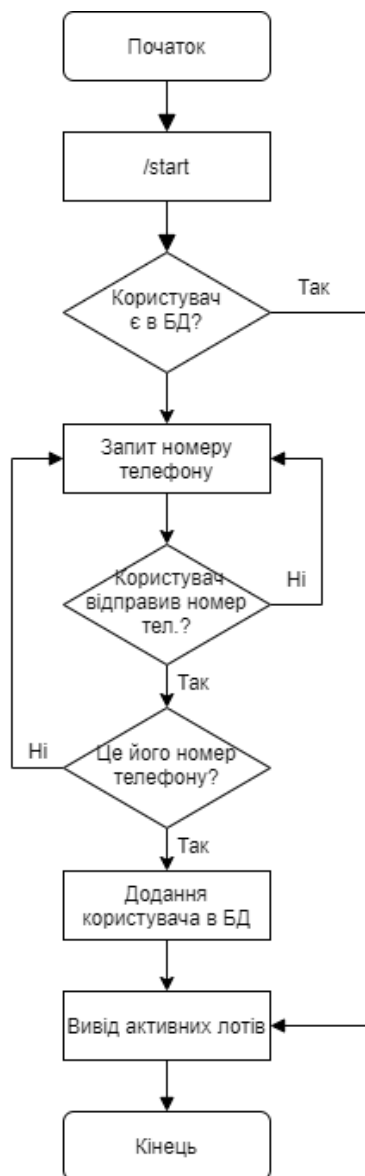
ДОДАТОК 2

Бот для проведення тендерів з продажу агрокультур

Схема функціональна
ІАЛЦ.467100.005 Д2

Аркушів 1

Київ — 2020



					ІАЛЦ.467100.005 Д2		
Зм.	Арк.	№ докум.	Підпис	Дата	Бот для проведення тендерів з продажу агрокультур Схема функціональна		
Розробив		Жуковський А.Л.					
Перевірів		Габінет А.В.					
Реценз.							
Н. Контр.		Сімоненко В.П.					
Затв.		Стіренко С.Г.					
					Літ. Аркуш Аркушів 1 1		
					НТУУ «КПІ», ФІОТ, ІО-64		

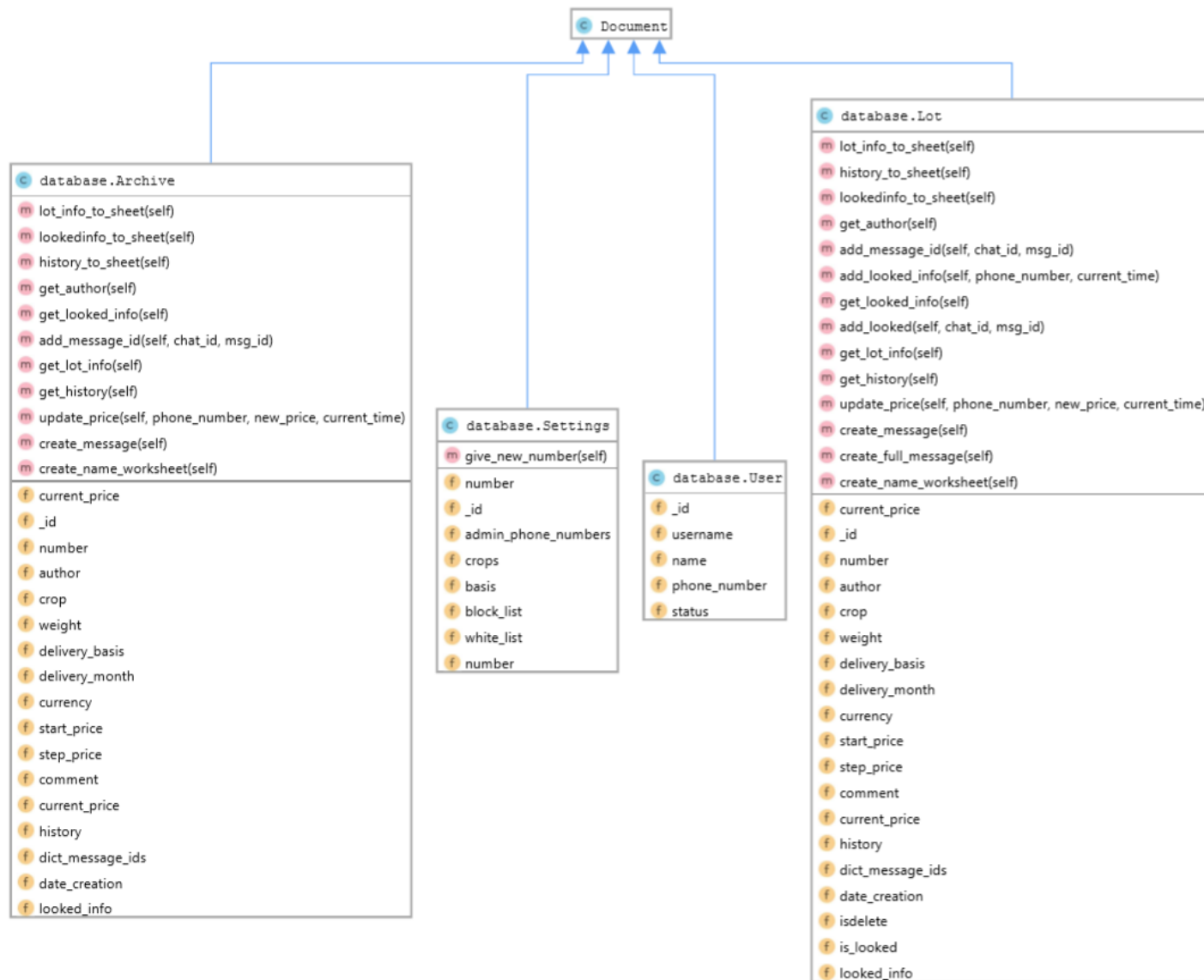
ДОДАТОК 3

Бот для проведення тендерів з продажу агрокультур

Схема принципова
ІАЛЦ.467100.006 ДЗ

Аркушів 1

Київ — 2020



					ІАЛЦ.467100.006 ДЗ									
Зм.	Арк.	№ докум.	Підпис	Дата										
Розробив		Жуковський А.Л.			Бот для проведення тендерів з продажу агрокультур Схема принципова				Лім.		Аркуш		Аркушів	
Перевішив		Габінет А.В.									1	1		
Реценз.														
Н. Контр.		Сімоненко В.П.							НТУУ «КПІ», ФІОТ, ІО-64					
Затв.		Стіренко С.Г.												